

★ Stefan Strobel • Thomas Uhl ★

LINUX

UNIVERSE



New
Release
1.2.X

EXTRA
MATERIALS
extras.springer.com

Linux Universe

Stefan Strobel Thomas Uhl

LINUX UNIVERSE

Installation and Configuration

with contributions by Rainer Maurer

Translation by Robert Bach



Springer Science+Business Media, LLC

Stefan Strobel
Schlegelstraße 19
D-74074 Heilbronn
Germany
stefan.strobel@linux.org

Thomas Uhl
Obere Heerbergstraße 17
D-97078 Würzburg
Germany
thomas.uhl@linux.org

Rainer Maurer
Jakob-Haspel-Str. 18
D-74078 Heilbronn
Germany

The authors can also be reached at the following e-mail address:
linux@hn-net.de

Further information about Linux Universe can be obtained at
<http://www.springer-ny.com/samples/linux/linux.html>

This work consists of a printed book and a CD-ROM packaged with the book, both of which are protected by federal copyright law and international treaty. The book may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA) except for brief excerpts in connection with reviews or scholarly analysis. For copyright information regarding the CD-ROM please consult the printed information packaged with the CD-ROM in the back of this publication. Use of the entire work in connection with any form of information storage and retrieval, electronic adaptation computer software or by similar or dissimilar methodology now known or hereafter developed other than those expressly granted in the CD-ROM copyright and disclaimer information is forbidden.

Portions of this text have been taken from Linux – Vom PC Zur Workstation, Second Edition, published by Springer-Verlag, Heidelberg.

Library of Congress Cataloging-in-Publication Data applied for.

Printed on acid-free paper.

© 1995 by Springer Science+Business Media New York
Originally published by Springer-Verlag New York, Inc in 1995.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Production managed by Ken Dreyhaupt; manufacturing supervised by Rhea Talbert.
Photocomposed copy prepared from the authors' Microsoft Word files.

9 8 7 6 5 4 3 2 1

Additional material to this book can be downloaded from <http://extras.springer.com>

ISBN 978-1-4757-2417-2 ISBN 978-1-4757-2415-8 (eBook)

DOI 10.1007/978-1-4757-2415-8

Acknowledgments

We wish to expressly convey our gratitude to the following persons, who actively contributed to the production of this book: Ruediger Hensch and the people from Unifix. Special thanks go to Rainer Maurer, who contributed the xadmin tool and its description in this booklet.

We also thank our translator, Bob Bach, for the synergy and the brutal night shifts that he shared with us in the final stages of preparing the manuscript for publication.

This CD could never have been produced without the preliminary work of Linus Torvalds and the international Linux network community that combined its energy to make Linux what it is today, and the pioneering efforts of Richard Stallman, founder of the Free Software Foundation, and the dedicated FSF network community that gave Linux a home in a broad base of software support.

Contents

Introduction..... 1

Features..... 3

Requirements

3.1 RAM..... 5
3.2 Hard disk 5
3.3 Bus system (ISA / EISA / PCI)..... 6
3.4 Video adapters 6
3.5 Peripheral hardware 6
3.6 PC and monitor 7
3.7 Supported hardware 7

Bootng Linux

4.1 Bootng under DOS..... 11
4.2 Bootng from a floppy disk 12
4.3 Installation via NFS 13
4.4 Bootng options..... 14

Installation

5.1 The keyboard.....	20
5.2 Partitioning.....	20
5.3 Swap.....	22
5.4 File system	22
5.5 X11	23
5.6 Install	27
5.7 Boot manager	27
5.8 Xadmin.....	29

Configuration

6.1 General configuration	31
6.2 Network configuration.....	39
6.3 X11 configuration.....	52
6.4 Configuration of X applications	63

Administration

7.1 The administrator	75
7.2 Shutdown.....	76
7.3 The directory tree	76
7.4 Users and groups	83
7.5 Shells.....	88
7.6 User information.....	89
7.7 Backups.....	89
7.8 Format and copy diskettes	90
7.9 File system management.....	91
7.10 Boot diskettes	93
7.11 Setting the system time	93
7.12 Linux Universe Packages.....	94

Reference	97
------------------------	-----------

Introduction

Congratulations on the purchase of your new Linux Universe CD. Welcome to the universe of Linux!

This manual is meant to help you install Linux on your computer and work with Linux. Linux Universe is preconfigured and delivered on CD-ROM for the utmost in ease of installation.

We aim to minimize your installation effort. Since you do not have to copy files to your hard disk, Linux Universe enables you to start working immediately. Nevertheless, all sources and documentation are available on-line at any time.

These instructions do not replace books on UNIX. This User's Manual only explains the most important operations regarding installation and provides you with necessary information. We assume that you are acquainted with computers and software and that UNIX is not totally new to you.

For more information on Linux and a more detailed overview of the system, please refer to our book:

Linux – Unleashing the Workstation in your PC

Stefan Strobel, Thomas Uhl

Springer Verlag

ISBN 3-540-58077-8

ISBN 0-387-58077-8

Features

Linux is a free UNIX-like operating system kernel. Our Linux Universe Distribution includes this kernel together with many tools from the Free Software Foundation's GNU project and other freely available programs and utilities. They were developed by many volunteers all over the world. Together they build a complete UNIX-like system.

To give you a better orientation, we offer the following summary of the most important features of Linux and our distribution:

- **A full-fledged 32-bit multi-user/multitasking UNIX system.** Linux permits multiple users to execute (different) programs simultaneously and thereby fully exploits the capacity of the Intel 80386 processor and its successors. The resulting performance is definitely comparable to a classical RISC workstation.
- **Orientation to common UNIX standards (POSIX).** Available software that adheres to existing UNIX standards usually can be ported to Linux without problems.
- **Network support (TCP/IP and others).** A machine running Linux Universe can easily be integrated into a TCP/IP, LanManger, Windows for Workgroups or Novell network. Linux supports common Ethernet adapters and TCP/IP connection via modem (SLIP, PPP).
- **Graphical user interface (X Window System).** The Linux Universe system includes the current version (Release 6) of the X Window System. OSF/Motif, the standard user interface for proprietary UNIX systems, can be purchased as an add-on product.

- **GNU utilities and programs.** Many of Linux Universe's commands and utilities emanated from the GNU project and contribute much functional enhancement.
- **Complete UNIX development environment.** Linux Universe permits the development of programs that run problem-free on other UNIX systems. In addition to the GNU C/C++/Objective C compiler, numerous editors, and several version control systems, there are numerous other software development tools.
- **Runs directly from the CD-ROM.** The Linux Universe System is preconfigured and runs directly from the CD-ROM. It needs only minimal space on your hard drive.
- **Smart CD-ROM cache.** The Linux Universe distribution profits from its smart CD-ROM cache to speed up access to the files on the CD-ROM. Accessed files are automatically copied to the hard-drive. The cache can be configured to allow maximum size of cached files.
- **Graphical administration tool.** The graphical tool `xadmin` supports configuration and administration of the system.
- **Complete ELF.** The entire Linux Universe distribution uses the new ELF link format, making it one of the first and most up to date distributions with ELF.
- **Compatibility to the iBCS2 Standard.** This allows the execution of programs in COFF and ELF format that were developed for SCO UNIX or other PC UNIX variants.

Requirements

The general prerequisite for running Linux is a PC with an 80386 or newer processor. Linux definitely does not run on old XTs or ATs with 80286 processors because Linux requires task-switching features that have only been available since the 80386.

80386 or newer

3.1 RAM

Working with the X Window System in the normal version requires at least 8 MB of RAM. If the machine is equipped with 16 MB, this delivers a noticeable improvement in performance, particularly under X11. Although in principle Linux runs starting with 4 MB of RAM, practical work at this level proves impossible.

8 MB or more RAM

3.2 Hard disk

You can use Linux Universe with a minimum of 20 MB of free storage space on your hard disk. To copy the most important parts of the system onto the hard disk and to use the Smart CD-ROM cache, we advise having approximately 100 MB.

100 MB HD

There is no upper limit to hard disk space. Anyone with access to the Internet can easily fill a 2 GB hard disk with Linux software. The availability of programming languages, utilities, libraries and application programs has become enormous.

no ceiling

3.3 Bus system (ISA / EISA / PCI)

Mother boards with the old AT bus (ISA), the more flexible and faster EISA bus, Local bus extensions, and the newer PCI bus are all supported.

PCI bus The PCI bus is processor-independent and significantly faster than the EISA system. It permits data transfer rates of up to 130 Mbytes/s in 32-bit mode. Some PCI boards have an extremely fast SCSI chip (NCR 53c810) for which the official kernel contains a driver.

NCR SCSI chip

Although the PCI standard was specified with utmost care, some PCI components available today deviate from it in part. Unfortunately, this can result in continued problems in the interplay between the mother board and peripheral boards. A dedicated PCI-HOWTO discusses such difficulties; you should certainly read it before you purchase PCI hardware.

problems

3.4 Video adapters

Linux principally supports all Hercules, EGA and VGA video adapters in text mode. However, to achieve higher resolution than 640 x 480 under X11 requires an appropriate X server. The XFree86 servers generally suffice to drive the most common PC video adapters under Linux. Particularly under X11, adapters with accelerator chips (S3, MachX) prove interesting. They enable speed that sometimes even exceeds the performance of professional graphic workstations.

X server

3.5 Peripheral hardware

streamer For data backup on streamers (tape drives) you can use a SCSI device or a cheaper floppy streamer, although we recommend the former.

network The selection of drivers for network adapters is quite broad. Beyond the popular adapters from Novell, 3Com and SMC, adapters by HP and D-Link are also supported. There are even drivers for Arcnet and token ring adapters. Numerous generic network adapters are compatible to these and can also be used.

In the UNIX tradition, with Linux you can use ASCII terminals connected to serial ports or specialized multi-serial adapters. There are also drivers for such adapters.

One feature that distinguishes Linux from many other systems is its direct kernel support of ISDN adapters. Drivers for Teles adapters reside on the usual Linux ftp servers. Drivers for adapters from other vendors are under development.

ISDN

Likewise, multimedia applications are possible under Linux. Peripherals such as sound boards (SoundBlaster, SoundBlaster 16, Adlib, Gravis Ultra Sound and PAS 16) and CD-ROM drives can be driven by corresponding kernel drivers. SCSI drives can also be installed without problems.

sound boards,
CD-ROM

3.6 PC and monitor

In view of continuing development in hardware and software, the new purchase of a 486 computer with at least 33 MHz and 16 MB RAM is a reasonable and affordable configuration. The hard disk should have at least 200 MB, since permanent shortage of disk space makes practical work impossible. If you intend to use a CD-ROM drive or a streamer in addition to your hard disk, you should take the SCSI path from the start.

486 + 16 MB

A 14 inch monitor proves too small for a multitasking environment because multiple windows tend to be open at once. However, the Linux X server can provide a large virtual monitor, and most devices can reasonably handle a resolution of 800 x 600 pixels; thus for starters you can do without a 17 inch monitor.

14" monitor too small

3.7 Supported hardware

Here we list the hardware currently supported by Linux. Not all the listed drivers are included in the compiled Linux Universe Distribution kernel. For rare peripheral devices, you might have to download the corresponding driver from an ftp server in the Internet and compile a new kernel.

Hard disk controllers:

- Generic XT (8 bit)
- IDE, MFM, RLL, ESDI (emulating ST-506)
- Enhanced IDE

SCSI host adapters:

- AMI Fast Disk VLB/EISA (BusLogic drivers)
- Adaptec 1505, 1515, 152x, 154x, 174x, 274x, 284x
- Always IN2000
- BusLogic (all models)
- DPT Smartcache (EATA)
- DTC 329x (Adaptec compatibility mode)
- Future Domain TMC-8xx, TMC-950, TMC-16x0, TMC-3260
- NCR 53c7x0, 53c8x0
- PAS 16 (ProAudioSpectrum/Studio16)
- Seagate ST-01/ST02
- SoundBlaster 16 SCSI-2 (Adaptec 152x)
- Trantor T128 and T228
- UltraStor 14F, 24F, 34F
- Western Digital 7000-FASST

CD-ROM drives:

- any SCSI CD-ROM (512, 2048 bytes block size)
- Mitsumi 8 and 16 Bit (150 or 300 KB/s)
- Matsushita, Panasonic, Kotobuki
- NEC CDR-260, CDR-35D
- SoundBlaster Pro, LaserMate Adapter
- Sony CDU 31A, CDU33A, CDU-535, CDU-531
- Philips CM 205/225/202

Graphic adapters with the following chipsets:

- Avance Logic AL2101
- ATI Mach8/32, 28800-4/5/a
- Cirrus Logic CLGD542x , CLGD6205/6215/6225/6235
- Compaq AVGA
- Genoa GVGA
- IBM 8514/A, XGA, XGA-II
- IIT AGX-010/014/015/016
- MCGA (320x200)
- MX MX68000/MX68010
- NCR 77C22/77C22E

- OAK OTI067/77
- Paradise PVGA1
- S3 801 / 805 / 864 / 911 / 924 / 928 / 964
- Trident TVGA8800CS / 8900 / 9000
- Tseng ET3000, ET4000, W32/W32i/W32p
- VGA (standard)
- Video 7 / Headland Technologies HT216-32
- WD WD90C00 / 10 / 11 / 30 / 31
- Weitek P9000

Network adapters:

- 3Com 3C503, 3C505, 3C507, 3C509 / 3C579
- AMD LANCE (79C960)
- Allied Telesis AT1700
- Ansel Communications AC3200 EISA
- Apricot Xen-II
- AT-Lan-Tec/RealTek parallel port adapter
- Cabletron E21xx
- DEC DEPCA and EtherWORKS
- D-Link DE600/DE620 parallel port adapter
- HP PCLAN / PCLAN PLUS
- Intel EtherExpress
- Novell NE1000 / 2000 / 2100,
- Racal-Interlan NI5210/ NI6510
- PureData PDUC8028, PDI8023
- SMC Ultra
- Schneider & Koch G16
- Western Digital WD80x3
- Zenith Z-Note

Sound cards:

- 6850 UART MIDI
- ATI Stereo F/X (SB compatible)
- Adlib
- ECHO-PSS (Orchid SW32, Cardinal DSP16)
- Ensoniq SoundScape
- Gravis Ultrasound, MAX
- Logitech SoundMan Games/Wave/16
- Microsoft Sound System (AD1848)
- MPU-401 MIDI
- Media Vision Premium 3D
- Pro Sonic / Pro Audio Spectrum-16

- SoundBlaster, SoundBlaster Pro, SoundBlaster 16
- Sound Galaxy NX Pro
- ThunderBoard
- WaveBlaster

PCMCIA cards:

- 3Com 3C589
- Accton EN2212 EtherCard
- D-Link DE650
- IBM Credit Card Adapter
- IC-Card
- Kingston KNE-PCM/M
- LANEED Ethernet
- Linksys EthernetCard
- Network General "Sniffer"
- Novell NE4100
- Thomas-Conrad Ethernet

Booting Linux

Linux Universe can be booted with a special diskette or from DOS. The system can be installed directly from the CD or via a network (NFS).

booting

4.1 Booting under DOS

This booting option requires that the corresponding CD-ROM driver for MS-DOS be installed to permit reading the Linux CD. In addition, sufficient RAM (ca. 500 KB) must be free to load the complete kernel.

RAM

The CD contains a program called `INSTALL.EXE`, which starts the installation procedure by loading the Linux kernel and then starting it. This program requires the type of the CD-ROM drive as parameter. The following types are supported:

INSTALL.EXE

sr0	first SCSI CD-ROM
hda	EIDE/ATAPI CD-ROM (master on first controller)
hdb	EIDE/ATAPI CD-ROM (slave on first controller)
hdc	EIDE/ATAPI CD-ROM (master on second controller)
hdd	EIDE/ATAPI CD-ROM (slave on second controller)
mcd	non-EIDE/non-ATAPI Mitsumi CD-ROM

sbpcd	Matsushita/Panasonic/Kotobuki CR-521/522/523/562/563 Longshine LCS-7260 IBM External ISA CD-ROM
sonycd	Sony CDU-31A/33A
cdu535	Sony CDU-535
lmscd	Laser Magnetic Storage (Philips) LMS-205
aztcd	Aztech CDA268-01A, Orchid CD-3110, Okano / Wearnes CDD110

I/O addresses Additional hardware options can be passed as additional parameters, for example to specify the interrupt or I/O address to be used by the driver. A detailed overview of possible parameters is given below.

During the boot phase warnings and error messages are displayed, but these can be ignored.

4.2 Booting from a floppy disk

RAWRITE.EXE In the root directory of the CD resides a program named RAWRITE.EXE, which supports writing a file to a diskette sector by sector. To create a boot diskette, it suffices to copy the disk image FD1440 onto a formatted diskette under DOS:

```
D:\>rawrite
RaWrite 1.2 - Write disk file to raw floppy diskette

Enter source file name: fd1440
Enter destination drive: a
Please insert a formatted diskette into drive A: and press -
ENTER- :
Number of sectors per track for this disk is 18
Writing image to drive A:. Press ^C to abort.
Track: 79 Head: 1 Sector: 16
Done.

D:\>
```

To boot Linux, leave this diskette in drive A (or return it there) and restart the computer.

installation After booting, the installation program automatically starts and displays the following question:

4.3 Installation via NFS

```
For manual installation, type Control-D to exit install menu.
Where is the Linux Universe CD-ROM?
Default devices: 1) sr0 2) mcd 3) sbpcd 4) lmscd 5) sonycd 6) cdu535 7) azt
8) hda 9) hdb 10) hdc 11) hdd 12) srl 13) nfs
Device name of CD-ROM 13 (<Enter> for automatic search):
```

To continue, enter the type of the CD-ROM drive from which installation is to run. An explanation of the types is given under "Booting under DOS" above.

4.3 Installation via NFS

Alternatively, Linux Universe can be installed over a network (NFS). Here the contents of the CD-ROM must be exported to any NFS server in the directory `/dist`. After booting from the diskette, device 13 (for NFS) must be selected in the installation script.

NFS server

```
For manual installation, type Control-D to exit install menu.
Where is the Linux Universe CD-ROM?
Default devices: 1) sr0 2) mcd 3) sbpcd 4) lmscd 5) sonycd 6) cdu535 7) azt
8) hda 9) hdb 10) hdc 11) hdd 12) srl 13) nfs
Device name of CD-ROM or <Enter> for automatic search: 13
```

Now specify the installation source as an IP address (e.g., 194.8.11.1) of the NFS server and the exported directory (`/dist`):

source IP address

```
Where is the Linux Universe CD?
NFS location: 194.8.11.1:/dist
```

In addition, specify the IP address of the target machine and its network mask:

```
What is your internet address?
internet address: 194.7.11.10
What is your Network mask?
Network mask [255.255.255.0]:
```

The remaining installation steps are identical for all installation variants.

4.4 Booting options

The options of various drivers can be passed to the kernel during booting. This primarily serves to configure interrupts and I/O addresses of the installed hardware during booting.

- **root=Device**
Specifies the device from which the kernel is to mount the root file system (e.g., `root=/dev/hda2`).
- **ro**
Defines that the root file system is to be mounted read-only.
- **rw**
Defines that the root file system is to be mounted read/write.
- **debug**
Sets the debug level within the kernel to 10.
- **no-hlt**
Disables invocation of the `hlt` command in the kernel's idle loop.
- **no387**
This option specifies that the kernel should use not the arithmetic coprocessor but instead the kernel's emulation, which must be compiled in the kernel.
- **reserve=IO address, length{, IO address, length ...}**
Prevents the device driver during booting from reading the specified range of I/O ports to independently find supported adapters.
- **ram disk=kilobytes**
Sets the size of the ramdisk to the specified value in kilobytes.
- **cram=Kilobytes**
A boot diskette can contain a compressed ramdisk at the end of the file system. This option specifies the start offset of such a ramdisk in kilobytes.
- **ether=IRQ, IO address, P1, P2, device**
Passes parameters of the installed Ethernet adapter to the corresponding driver in the kernel. The meaning of P1 and P2 depends on the respective driver. Normally 0 is specified

here. `device` contains the name of the Ethernet device to which the parameters apply. The default is `eth0`.

- **`hdx=cylinders,heads,sectors`**

Defines the geometry of an IDE/EIDE device (hard disk or CD-ROM). This option is only needed if the device was not properly recognized. The following values are possible for `hdx`:

<code>hda</code>	master of 1st IDE/EIDE interface
<code>hdb</code>	slave of 1st IDE/EIDE interface
<code>hdc</code>	master of 2nd IDE/EIDE interface
<code>hdd</code>	slave of 2nd IDE/EIDE interface

`hdd=noprobe` suppresses seeking a slave on the second interface.

`hdc=cdrom` defines the device on the second master IDE interface as a CD-ROM drive.

- **`st=bufferSize,WT,MaxTapeBuffer`**

Specifies parameters for a SCSI tape drive. The exact meaning of the parameters can be found in the file `README.st` in the directory `drivers/scsi` of the kernel source code.

- **`bmouse=IRQ`**

Specifies the IRQ of the bus mouse.

- **`max_scsi_luns=Nr`**

Specifies the highest logical unit number for SCSI devices.

- **`floppy=XXX`**

Configures the floppy disk driver.

`floppy=all_drives` induces a search for more than two drives.

`floppy=daring` forces maximum-speed access to the floppy drives, which is not supported by all controllers.

`floppy=two_fdc` searches for a second floppy disk controller.

`floppy=0x370,two_fdc` specifies the address of the second controller.

`floppy=thinkpad` informs the floppy driver that the floppy drive uses inverted disk-changed line.

`floppy=2,5,cmos` informs the driver that the third floppy drive (#2) is an ED drive. The following types are possible:

Type	Designation
0	unknown
1	5 ¼ DD drive
2	5 ¼ HD drive
3	3 ½ DD drive
4	3 ½ HD drive
5	3 ½ ED drive

`floppy=no_unexpected_interrupts` suppresses the reporting of unexpected interrupts.

- **st0x=ROM address, IRQ**
Sets parameters for Seagate st01 and st02 host adapters.
- **tmc8xx=ROM-address, IRQ, sectors, heads, cylinders**
Sets parameters for Future Domain TMC8xx host adapter.
- **tmc16x0=ROM address, IRQ, sectors, heads, cylinders**
Sets parameters for Future Domain TMC16X0/3260 host adapter.
- **t128=ROM address, IRQ**
Sets parameters for T128 host adapter.
- **pas16=IO address, IRQ**
Sets parameters for PAS 16 host adapter.
- **ncr5380=IO address, IRQ, DMA**
Defines parameters for a SCSI host adapter with NCR 5380 chip.
- **us14f=IO address**
Sets the IO port address for UltraStor 14F / 34F SCSI host adapter.
- **aha152x=IO address, IRQ, SCSI-Id, reconnect, parity, debug**
Sets parameters for Adaptec 152x controller. Reconnect, parity and debug are Boolean flags (0=off, 1=on).
- **aha1542=IO address, bus-on interval, bus-off interval, DMA speed**

Sets parameters for Adaptec 154x-host adapter. Valid values for the bus-on interval are 2-15 μ s, for the bus-off interval 1-64 μ s, and for DMA transfer 5-10 MB/s.

- **aic7xxx=extend**

Requests extended BIOS geometry translation.

- **buslogic=IO address**

Sets parameters for Buslogic SCSI adapter.

- **mcd=IO address, IRQ, Mitsumi_bug_wait**

Sets parameters for Mitsumi CD-ROM drives.

- **sound=0xTaaaId**

Sets parameters for sound boards. Parameters are passed in a single hexadecimal number where the individual digits mean:

T type of board, with the following options:

- 1 FM Synth (YM3812 or OPL3)
- 2 SoundBlaster 1.0 to 2.0, SoundBlaster Pro and 16
- 3 Pro Audio Spectrum 16
- 4 Gravis UltraSound
- 5 MPU-401 UART Midi
- 6 SB16 with 16-bit DMA number
- 7 SB16 Midi (in MPU-401 emulation)

aaa IO address

I IRQ

d DMA channel (0,1,3,5,6 or 7)

- **pcsp=sampling rate**

Specifies the sampling rate of PC speaker device.

- **sbpcd=IO address, type**

Sets parameters for SoundBlaster / Panasonic CD-ROM driver. Type can be one of the following:

LaserMaster	SB CD on SB-compatible controller
SoundBlaster	SB Pro CD-interface
SPEA	SPEA FX / Sequonia S-1000

- **cdu31a=IO address, IRQ**

Sets parameters for Sony CDU-31A CD-ROM driver.

- **sony535=IO address**

Sets I/O port address for Sony CDU-535 CD-ROMs.

- **aztcd=IO address, 0x79**

Defines port address for Aztech, Orchid or Okano CD-ROM

drives. The optional `0x70` parameter forces the use of the drive even if the firmware test failed.

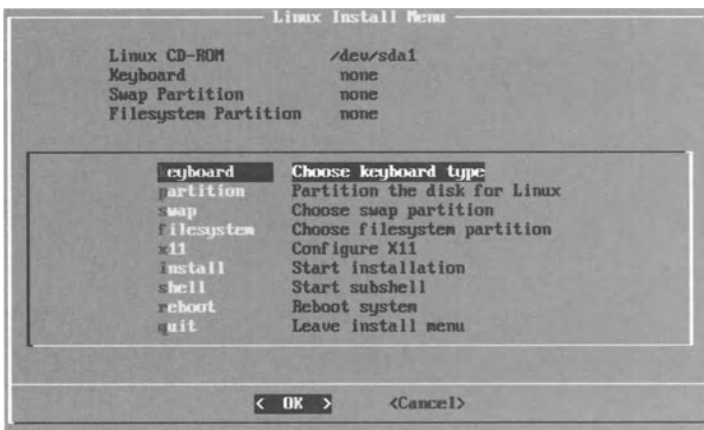
- **`lmscd=IO address, IRQ`**

Determines I/O address and IRQ for Laser Magnetic (Philips) CM-205 drive.

kernel All other options are not interpreted by the kernel, but passed on to the `init` process. If the option contains an equal sign (=), then it is passed as an environment variable, otherwise as a command line parameter.

Installation

After successfully booting Linux, you can begin the actual installation. The installation program starts automatically and the Linux Universe installation menu appears.

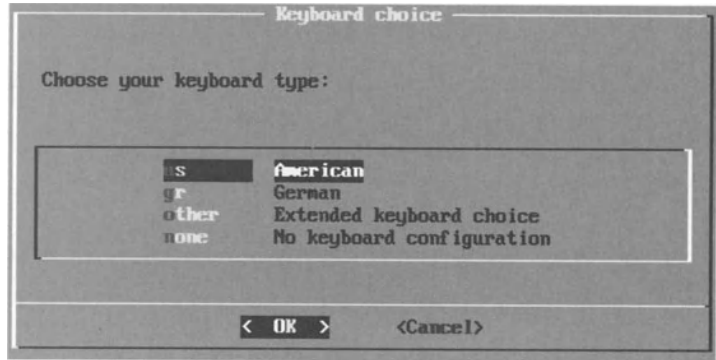


Main menu

The top four lines display status information, including the path of the installation CD, the current keyboard layout, the path of the swap partition, and the selected installation path. status information

5.1 The keyboard

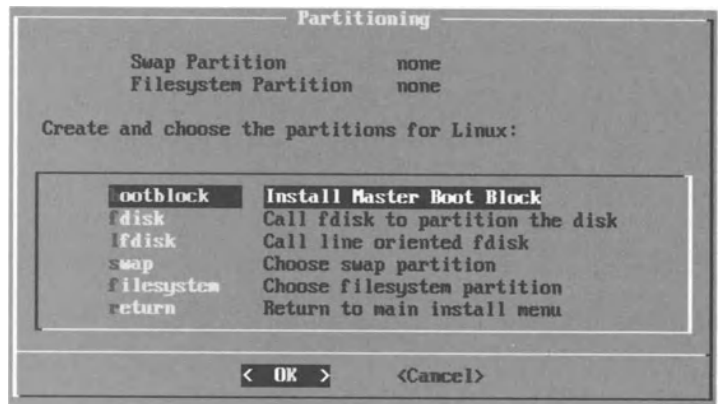
Selecting the first entry in the main menu invokes a new submenu. The cursor keys and <Return> permit selection of the national keyboard layout.



Keyboard layout

5.2 Partitioning

Installation requires two free partitions; these are created via the *Partition* submenu.

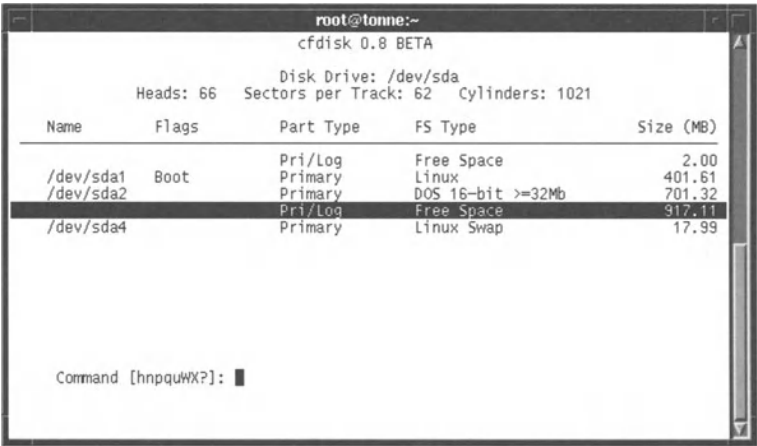


Partition menu

The installation of a special master boot record (MBR) is only necessary if Linux will not be loaded from the first hard disk. If Linux is loaded from the first hard disk, then booting can be executed with the MS-DOS master boot record, which is already present on the hard disk if an MS-DOS system has been installed. For partitioning, select the hard disk drive from the submenu. The designations of the hard disks are given in the following table:

Device	Designation
/dev/hda	1st IDE/EIDE hard disk
/dev/hdb	2nd IDE/EIDE hard disk
/dev/sda	1st SCSI hard disk
/dev/sdb	2nd SCSI hard disk

Two programs are available for partitioning, `fdisk` and `lfdisk`, but they differ only in their interface. Since it is more comfortable, we describe only the former version.



fdisk

Before creating a new partition, you need to select the entry *Free Space* with the cursor keys. New partitions are created with the command `n`.

Then choose whether you want to create a *primary* or a *logical* partition. As under DOS, the number of primary partitions is limited to four.

The installation of Linux Universe requires two free partitions. The first should be at least 100 MB. This is where the Linux kernel, important system programs and user data will reside. Type 83 (Linux) is automatically selected for this partition.

swap partition

The second partition is needed for swapping and should be at least 8 MB or at least the size of your machine's RAM, whichever is larger. This partition is type 82 (Linux Swap). A swap partition is activated later via the main menu.

boot partition

The Linux data partition should be used as boot partition. Other operating systems can then be started via the Linux Universe boot manager. The command `` sets or resets the boot flag of the selected partition.

Current settings are written to the hard disk with `<w>`. Then the computer needs to be restarted. Thus an appropriate dialog is displayed. With `<q>` or `<Q>` the program can be terminated without activating the modifications. All further commands are displayed on input of `"?"`.

5.3 Swap

swap partition

For the creation of virtual memory, Linux requires a separate partition (swap partition), which is created with `fdisk` (see *Partitioning* above). The swap partition can be selected on booting in the main menu or in the submenu *Partitioning*, which offers all partitions of type 82 (Linux swap). Selecting *Start Swapping now* initializes the swap partition and activates swapping.

5.4 File system

installation partition

Menu item *File System* in the main menu provides a selection of Linux installation partitions. If there are data on the selected partition, these can be copied into a directory named `/old`.

5.5 X11

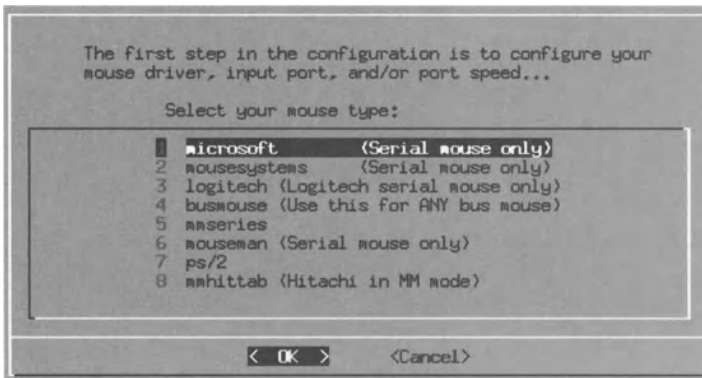
To configure the X11 system, select item *X11* from the main menu. After some seconds the configuration utility appears:



X11 configuration menu

First check the mouse settings:

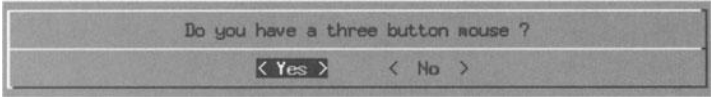
mouse



Selection of mouse type

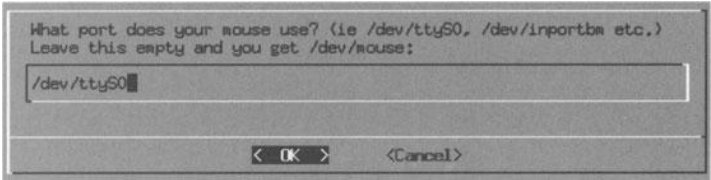
Most serial mice are Microsoft-compatible. Hence in the following we handle only this type further. If you have a three-button mouse, you must specify this explicitly:

Microsoft mouse



Mouse button configuration

path In addition, we need the path of the mouse port.

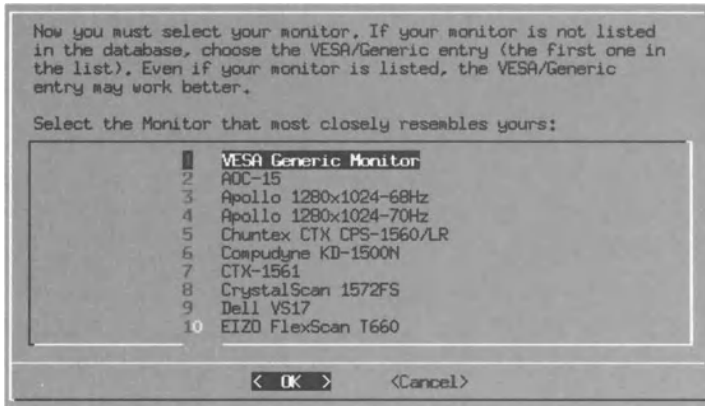


Specifying the mouse port

The following mouse types and ports are supported:

Path	Mouse type
/dev/ttyS0	serial mouse on COM1
/dev/ttyS1	serial mouse an COM2
/dev/bmouseps2	PS/2 bus mouse
/dev/bmousems	Microsoft bus mouse
/dev/bmouselogitec	Logitech bus mouse
/dev/bmouseatixl	ATI bus mouse

Next comes the monitor:



Selection of the monitor

Take extreme care to select the correct monitor, since your monitor might otherwise be damaged or destroyed. If you cannot find your monitor in the list, use the *VESA generic monitor*. Likewise the video adapter must be configured carefully:

Caution!

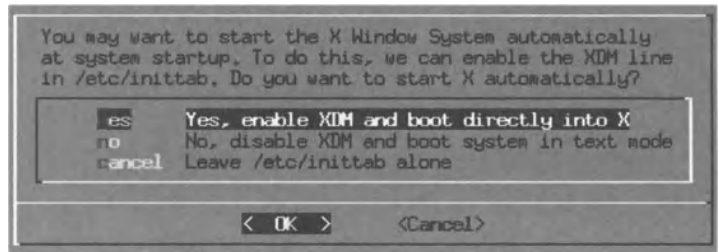


Selection of video adapter

Use generic types only if you cannot find your video adapter in the list. The generic types generally permit only low resolutions. The generic VGA server (max. 16 colors) should work with all VGA-compatible adapters.

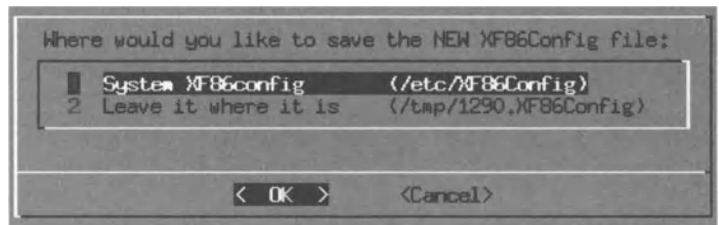
generics

On completion of the hardware configuration, all settings should be checked before continuing the installation. Select the menu item *Start X Server for a test*. After some seconds a black-and-white pattern will appear and the mouse cursor can be moved across the screen. If this is not the case, then there is an error in the hardware configuration. Simultaneously pressing the keys <Ctrl><Alt><Backspace> terminates the X server and returns to the main menu to allow correction of the erroneous settings. If a graphical login is desired after booting (XDM), this feature can be activated with menu item 6.



XDM startup

save Once the settings have been tested successfully, they can be saved in /etc/XF86Config:



Saving the configuration

This completes the configuration of the X Window system. The X Window setup can be reinvoked at any time with the command `ConfigXF86`, even if the complete system has already been installed.

5.6 Install

The main menu entry *Install* starts the actual installation procedure. After an appropriate request for confirmation, a new file system is created on the selected Linux partition, which can take some minutes. Then the installation program copies about 10 MB from the CD-ROM onto the hard disk and installs the boot manager. Most of the Linux Universe Distribution is left only on the CD-ROM or the NFS server and is loaded from there on demand.

After successful installation of the minimum base system, you need to execute a **reboot** from the main menu. The system responds with a login prompt. You can now log in as guest or as administrator (`root`).

5.7 Boot manager

The Linux Universe boot manager enables easy selection of the operating system to be booted.

Linux Universe --	
Name	Location
floppy	@:/vmlinuz rw ram disk=1440 cram=800
Linux hardware options	
	+-Keyboard+-Timeout-
	us 5
	+-+--
bootmanager - boot Choice Mode	
Up/Down: Choose Enter: boot F4: Edit F%: Enter Command Line Mode	

In the left top column the names of available boot configurations appear. The right column contains the corresponding boot parameters. A Boot configuration can be selected with the cursor keys and activated with `<Return>`.

On booting, if no key is pressed during a configurable timeout,

boot-configuration

the boot manager automatically executes a default configuration. General kernel parameters that apply to all configurations can be entered in the middle window.

Configuration

- edit mode

<F4> switches to edit mode, which allows creation of new boot configurations and modification of existing ones. Select the entry to be modified with the cursor keys or <Tab>. Pressing <F4> again opens a small dialog window where the desired modifications can be made.
- boot timeout

The duration of the boot timeout can be configured in the right bottom window. Also, the national keyboard layout can be modified. <F5> defines the current configuration as the default boot configuration. <F3> saves the settings and terminates edit mode.

The hard disk drive and the partition must be specified for every boot configuration. The hard disk is designated by a letter and the partition by a number. Thus to boot from the second partition of the first hard disk, it suffices to specify C:2.
- kernel path

Booting Linux also requires specification of the desired kernel. Furthermore, any number of additional parameters can be appended. A useful example configuration might be the following:

Designation	Parameters
multiuser	C:2/vmlinuz ro root=/dev/hda2
single user	C:2/vmlinuz ro root=/dev/hda2 single
DOS	C:1

- single-user mode

The first configuration starts the kernel `vmlinuz`, located on the second partition of the first hard disk. The root file system `/dev/hda2` also resides on the second partition. The additional parameter in the second variant assures that the system starts in single-user mode. The third variant boots MS-DOS on the first

partition. Additional kernel boot options can be found in the overview at the end of the previous chapter.

Master boot record

If Linux is to boot from the second hard disk, then the Linux Universe master boot record must be written to the first hard disk on installation. Any pre-existing DOS MBR is overwritten.

The new MBR responds to the first start with a primitive menu:

DOS MBR

```
C: X 1234 ABCD PQS ? _
```

The first letter (C:) indicates the drive from which the MBR was loaded. The hard disk is selected with a letter (usually C or D) and the boot partition with a number. The letter x continues booting.

To keep from having to make this selection on each booting, entering p (persistent) stores a selection. This invokes a second prompt. Here the boot drive and the partition can be specified again, but this configuration remains persistent.

To make modifications, the user can interrupt the booting process by entering <ESC> if needed. The time span in which this interrupt is possible can be adjusted with the keys q (quicker) and s (slower).

escape

5.8 Xadmin

After successfully installing Linux Universe, log yourself in as superuser (root) and invoke the graphical interface with startx. In the icon bar at the right is a button named *Xadmin*, which starts the graphical administration and configuration program. New users can be created and the most important system files can be modified. The next chapter provides more information on using this program.

Configuration

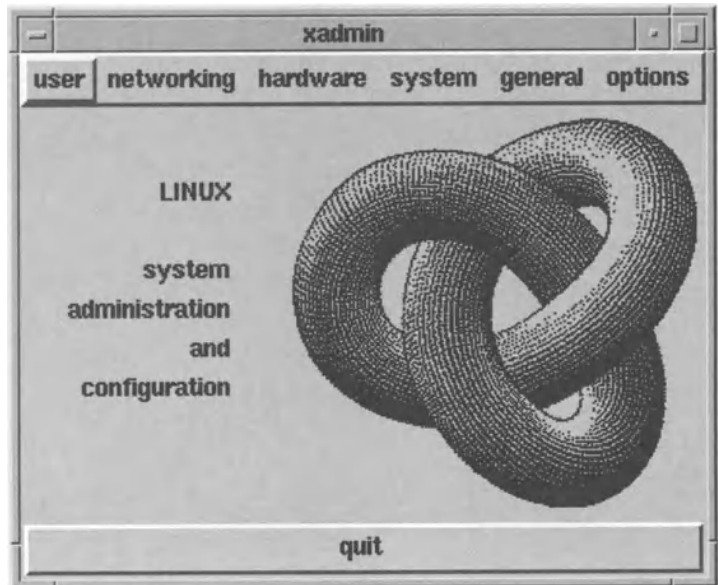
Once the system files have been copied onto the hard disk and the Linux Loader has been installed, some of the configuration files need to be adapted. This fine tunes the operating system to the available hardware.

6.1 General configuration

Most configuration files belong in the directory `/etc`. Many of these files are fixed in content and normally do not need to be changed. The appendix contains a brief description of these files. This chapter discusses the configuration modifications that have to be carried out after the installation of Linux on normal PC hardware.

configuration files

We recommend using the program `xadmin`, which is included in the Linux Universe distribution to help you configure and administrate your Linux system.



xadmin

`xadmin` enables you to set basic administration and configuration parameters without the need to find the corresponding ASCII files and edit them by hand with a conventional text editor. It displays the current settings of your file system table, your modem, and so on in a very readable manner and aids you in changing the settings by providing you with possible values and additional information about specific settings.

Many parameters can be set just by using your mouse, without any keyboard action. No changes are performed until you explicitly confirm the new settings. You can always cancel any module of `xadmin` without applying your changes.

General notes on the usage of `xadmin`

When you start `xadmin`, it displays a menu in the top containing the menu buttons "user", "networking", "hardware", "system", "general" and "options". At the bottom, `xadmin` displays a button that allows you to leave the program. You can start `xadmin`'s modules by clicking on a menu button and releasing it over the menu entry of interest (e.g., general issue).

All modules have a status line at the bottom. The information displayed in the status line depends on the location of the mouse pointer and describes the function of buttons or the meaning of specific configuration settings.

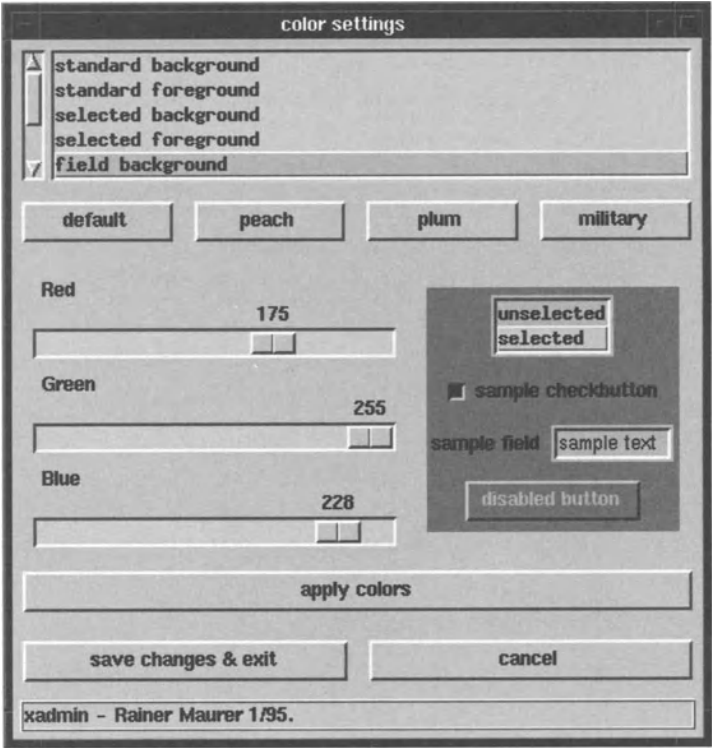
Right over the status line you will usually find the buttons "save changes & exit" (or sometimes "perform changes & exit") and "cancel". By pressing the button "cancel" you can leave a module any time without applying any changes. The settings will be the same as they were before you entered the module. By pressing the button "save changes & exit" you tell `xadmin` to apply your settings changes before the module is closed. The upper area of the modules allows you to edit the module-specific settings.

Some global settings for `xadmin` can be set in the sub menu "options". The module "defaults" allows you to configure settings like the default shell for new users. The module "colors" enables you to personalize `xadmin`'s colors.

status line

cancel

save changes
and exit



color configuration in xadmin

File systems

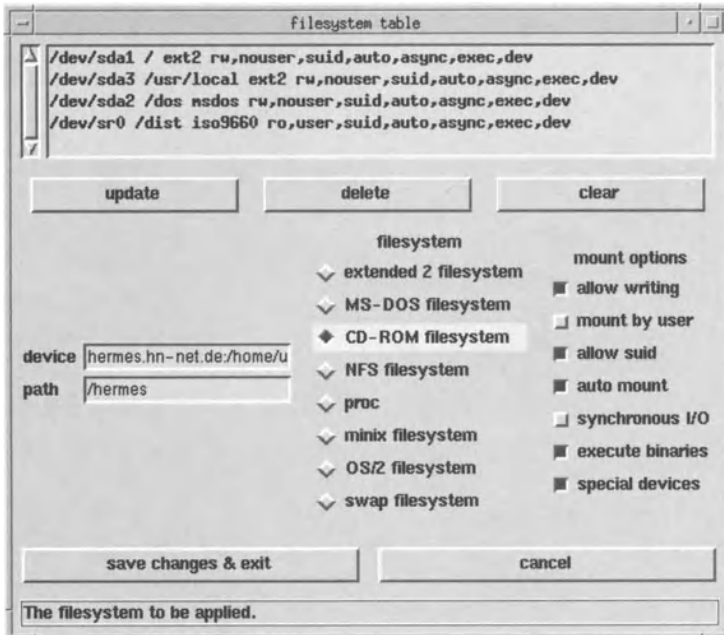
- mounting
- /etc/fstab

multiple file systems
- When the system starts up, one of the `rc` scripts mounts all file systems specified in the file `/etc/fstab` into the directory tree. This file enumerates all available file systems along with their options, which are passed as parameters to the `mount` command. If multiple file systems have been created, these should be entered in the file `/etc/fstab`. The following example (excerpt from the file `/etc/fstab`) shows these entries for a computer with multiple file systems:

/dev/sda2	/	ext2	defaults
/dev/sda5	/home	ext2	defaults
none	/proc	proc	defaults
/dev/sdb1	/mnt1	ext2	defaults,noauto
/dev/sda3	none	swap	

The entry that mounts the `/proc` file system is important because several commands, such as a variant of `ps` and `xadmin`'s module processes, are dependent on this directory. System information is represented here in the form of files and subdirectories. The last entry is not a file system but a swap partition. This is described in the next section.

The file system module in `xadmin` reads and writes the file `/etc/fstab`. For each of your disk devices you can specify the corresponding mount path, file system and options. The list at the top contains the current file system table settings in the way they will be written to `/etc/fstab` if you press the button "save changes & exit".



The file system module of `xadmin`

If the parameter `noauto` is used as an option in the file `/etc/fstab`, the specified file system is not mounted automatically when the system boots but has to be mounted explicitly. This is used for removable media or special NFS mounts. For a complete description of the options, see the `Manual` page for the `mount` command.

To edit a line of the list, click on it with the left mouse button. You may now edit the entries "device" and "path" and select a file system and the mount options. The button "update" changes the settings for the device in the list or adds it if it is not in the list. The button "delete" removes the selected line from the list. Pressing the button "clear" initializes the device, the path, the file system and the mount options.

Swapspace

RAM If your computer is equipped with 8 MB or less of RAM, you will need to set up a swap partition or a swap file, unless you have already done so during installation. Otherwise your memory will not suffice to run multiple programs simultaneously under the graphical user interface. 4 MB does not even suffice to recompile the kernel while an editor runs alongside. Even with 16 MB it makes sense to enhance virtual memory with a swap partition.

swapon The swapspace is activated with the command `swapon`. To induce this automatically at system startup, enter the swap partition like a file system in the file `/etc/fstab`. The example above already contains such an entry.

swap file Sometimes it makes sense to use a swap file instead of a swap partition. Such a file must first be created in the desired size, which can be done with the command `dd`. The following example creates and activates a swap file:

```

hermes:/# free
             total        used        free      shared    buffers
Mem:         31380        29476        1904       17260       10092
Swap:         0            0            0
hermes:/# dd if=/dev/zero of=/swapfile bs=1k count=8192
8192+0 records in
8192+0 records out
hermes:/# mkswap /swapfile 8192
Setting up swapspace, size = 8384512 bytes
hermes:/# sync
hermes:/# swapon /swapfile
hermes:/# free
             total        used        free      shared    buffers
Mem:         31380        29484        1896       17260       10092
Swap:         8188            0        8188
hermes:/#

```

Swap files can also be entered in the file `/etc/fstab` so that they are automatically activated on system startup with `swapon -a`. Here we enter the swap file instead of the device file:

```

/swapfile    none    swap

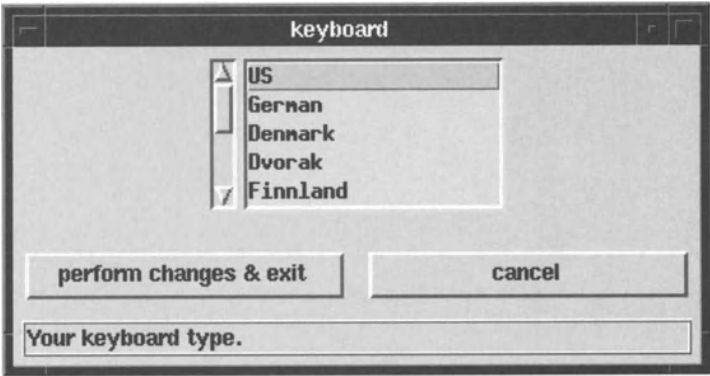
```

Likewise in the file system module of `xadmin` the swapspace can be entered like a file system.

Keyboard layout configuration

Since Version 0.99.10 of the Linux kernel, keyboard layout is no longer bound to the compilation of the kernel, but can be configured at run time. Contrary to other Linux distributions, which configure the keyboard separately for the console and the graphical user interface, Linux Universe allows these configurations to take place together.

The configuration is stored in the file `/etc/KEYBOARD` and can be modified with the command `keyboard` or `xadmin`. Naturally the file can be modified directly with an editor as well.

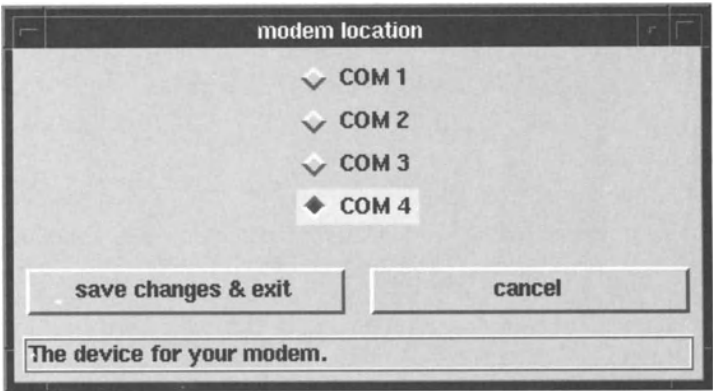


setting the keyboard layout with xadmin

Modems

COM1 through
COM4

The `xadmin` module "modem" allows you to specify your modem device. The ports "COM1" through "COM4" may be selected by clicking on the appropriate radio button. Pressing the button "save changes & exit" will create a link `/dev/modem` to the respective device (`/dev/cua0`, ..., `/dev/cua3`).



setting the modem port

6.2 Network configuration

Anyone with access to a TCP/IP network, whether through a university with Internet connection, at a company, or at home, can connect a Linux PC to this network. The following subsections describe the procedure in detail.

Addresses

In a network connection with TCP/IP, each network interface of a computer receives an IP address that is unique worldwide. The address consists of four numbers between 0 and 255, separated by periods, for example, 141.7.1.40.

Other important data necessary for the configuration of the network are the network mask, broadcast address, the address of the router, and possibly the address of a name server. These terms are explained briefly in the following subsections. However, a detailed discussion of the administrative details would be beyond the scope of this book. Thus in the following we deal only with the practical effects of these parameters and refer the interested reader to the many books dedicated to TCP/IP and network administration.

The exact definition of the protocols and concepts can be taken from the RFCs (Requests for Comments), which can be downloaded as text files from any larger ftp server.

Networks usually have a network administrator who manages the addresses and knows all important parameters. For small private networks that lack connections to other networks, the IP address is relatively insignificant. The user(s) can select them at will or assume the default values suggested by the installation program. Here it only is important to ensure unambiguity within the local area network.

When a direct connection to the Internet is being set up, the machine is assigned a network address in one of the classes A, B, or C. Depending on its class, this network address consists of one, two, or three bytes. The first bits of an address indicate the class to which the address belongs.

TCP/IP

IP address

network mask,
router, nameserver

RFCs

network classes

The network address constitutes the start of the IP address. The local network administrator can assign the remaining bits. In a class C network the IP addresses consist of an officially determined network address with 3 bytes and a local component with 1 byte. Since addresses ending with 0 or 255 have a special meaning, a class C network can encompass up to 254 IP addresses.

subnetworks The local component of the IP address can either be used directly for addressing the individual machine or divided into a subnetwork address and a host address by means of a network mask. However, we do not further discuss the use of subnetworks here. Details can be found in rfc950.

network mask In the following let us assume a class C network without subnetworks. The network mask for such a network is 255.255.255.0. This means that 3 bytes are used for network addressing and the remaining byte for host addressing.

The network address is entered with a trailing zero. In our example the network address might be 193.48.121.0 and the IP address of the Linux machine 193.48.121.37.

broadcast address Another parameter needed for the configuration of the network is the broadcast address. TCP/IP uses a broadcast to handle certain tasks. In the simplest case and in our example, this is the network address with a final 255 instead of 0, i.e., 193.48.121.255.

For other computers on this network, only the last number of the IP address should be changed and only numbers from 2 to 253 should be used. Number 1 or 254 is normally used for a router and 0 and 255 for the network and broadcast addresses, respectively.

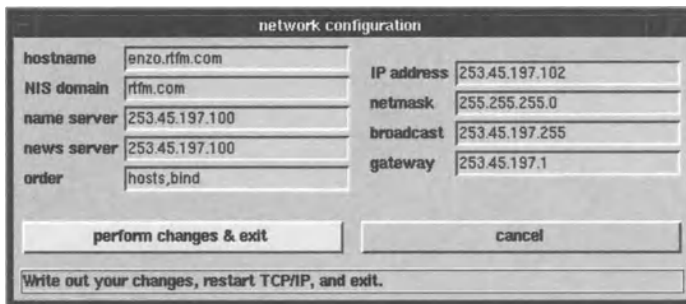
Loopback device

virtual interface Even for computers that do not actually possess a network interface, there is a virtual interface named *loopback*. As its name suggests, everything that is output via this interface is input again directly. This allows TCP/IP connections from a machine to itself. Thus the TCP/IP programs can be used even without a

network interface. The IP address of the loopback interface is usually 127.0.0.1.

Configuration of network interfaces

With Linux Universe you can completely configure the network with the graphical tool `xadmin`. It suffices to enter the correct values in the mask. They are then stored in the file `/etc/NETCONF` and evaluated upon leaving the network module `NETCONF` in `xadmin` and on booting.



`xadmin`'s TCP/IP module

The input field *hostname* specifies the name of the user's machine along with the respective domain name. *NIS domain* contains the name of the NIS domain (formerly Yellow Pages), but can be omitted if NIS is not used.

The field *name server* should contain the IP address of the name server, if available (see below). The field *news server* can contain the name of a news server, which is then used by programs for reading the news from Usenet.

The field *order* is used for the file `/etc/host.conf` and specifies the order in which the resolver (see below) attempts to resolve symbolic host names.

The subsequent fields are needed to configure the network interface and were already explained above. *Gateway* should contain the name or address of the router if one is present.

Host names and name servers

Most users do not consider the direct input of an IP address to be a very comfortable solution. This motivated the introduction of symbolic addresses. Such a symbolic address consists of the host name and the domain name.

host name,
domain name

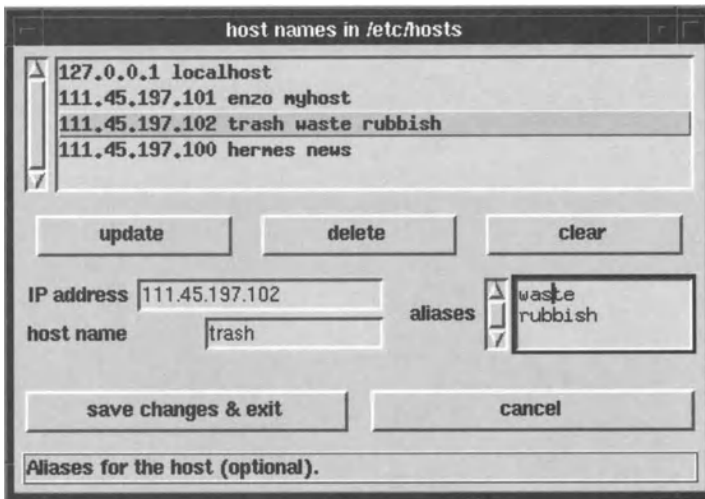
The host name is a locally unique designation for a machine. The domain name, on the other hand, must be internationally unambiguous. Instead of the IP address 152.2.22.81, for example, a user could enter the symbolic address sunsite.unc.edu. In this case sunsite is the host name and unc.edu the name of the domain.

alias

The file /etc/hosts establishes the one-to-one correspondence locally of IP address and symbolic address. Here additional names, or aliases, can be defined for a host. Aliases serve to define shorter names and thus to make the entry of addresses more comfortable. The following is an excerpt from the file /etc/hosts:

#IP address	symbolic address	alias	
141.7.21.40	linux1.test.fh-heilbronn.de	linux1	
127.0.0.1	localhost		
141.7.21.20	sun1.test.fh-heilbronn.de	sun1	
141.7.21.25	risc1.test.fh-heilbronn.de	risc1	news

This file can also be maintained with xadmin:



/etc/hosts with xadmin

To add a host, insert the corresponding information in the entry fields "IP address", "host name" and "aliases" and press the button "update". The button "update" will change a line in the list at the top if the IP address already exists or add a new host, respectively. To select a host in order to edit or delete it, click on the appropriate line in the list with the left mouse button. The button "delete" will remove the selected line from the list. You may clear the information displayed in the entry fields by pressing the button "clear". To leave the module, press the button "save changes & exit" if you want to write out a new `/etc/hosts` or the button "cancel" if you don't want your changes to be applied.

Since maintaining hundreds or thousands of entries in this file proved impossible, a hierarchical system of name servers was set up to manage these symbolic names and automatically exchange addresses among themselves. For each domain, i.e., all machines with a certain domain name, there is a responsible name server. A detailed description of the concepts domain name, subdomain, and name server can be found in `rfc1034` and `rfc1035`.

The UNIX TCP/IP programs convert the host name to the corresponding IP address by invoking a C library routine. This routine, called *resolver*, reads the file `/etc/hosts` and makes

a connection to the next name server if necessary. The order in which this is to occur can be specified with `xadmin's` module "tcpip". The information is stored in the file `/etc/host.conf`.

Inetd

Internet The `inetd` daemon manages Internet services. It waits for requests for connections to port numbers that have been specified for Internet services and starts the corresponding daemon only upon an actual request for a connection. The file `/etc/inetd.conf` contains the table that specifies which daemon is responsible for which services.

telnet	stream	tcp	nowait	root	/etc/telnetd	telnetd
ntalk	dgram	udp	wait	root	/etc/ntalkd	ntalkd
ftp	stream	tcp	nowait	root	/etc/ftpd	ftpd -l
finger	stream	tcp	nowait	root	/etc/fingerd	finger
shell	stream	tcp	nowait	root	/etc/rshd	rshd
login	stream	tcp	nowait	root	/etc/rlogind	rlogind
tftp	dgram	udp	wait	root	/etc/tftpd	tftpd echo
echo	dgram	udp	wait	root	internal	
discard	stream	tcp	nowait	root	internal	
discard	dgram	udp	wait	root	internal	
daytime	stream	tcp	nowait	root	internal	
daytime	dgram	udp	wait	root	internal	
chargen	stream	tcp	nowait	root	internal	
chargen	dgram	udp	wait	root	internal	

port numbers The file `/etc/services` establishes the correspondence of services to port numbers.

tcpmux	1/tcp	# TCP Port Service Multiplexer
rje	5/tcp	# remote job entry
echo	7/tcp	
echo	7/udp	
discard	9/tcp	sink null
discard	9/udp	sink null
systat	11/udp	users
systat	11/tcp	users
daytime	13/udp	
daytime	13/tcp	
daytime	13/udp	
netstat	15/udp	
netstat	15/tcp	
gotd	17/udp	quote
quote	17/tcp	# quote of the day
chargen	19/tcp	ttytst source
chargen	19/udp	ttytst source
ftp data	20/tcp	
ftp	21/tcp	
telnet	23/tcp	
smtp	25/tcp	mail #Simple Mail Transfer
nsw-fe	27/tcp	# NSW User System FE [24, RHT]

These files normally remain unchanged. Their modification becomes necessary only when new services are added or when the update of a daemon involves new options.

Berkeley r-Utilities

The Berkeley r-Utilities `rlogin`, `rsh` and `rcp` also have their own configuration files. In order to be granted remote access to another computer with these programs, the client machine must be entered in the file `/etc/hosts.equiv` on the server. A user name can be entered along with the host name in this file. This restricts access to this specific user on the specified host.

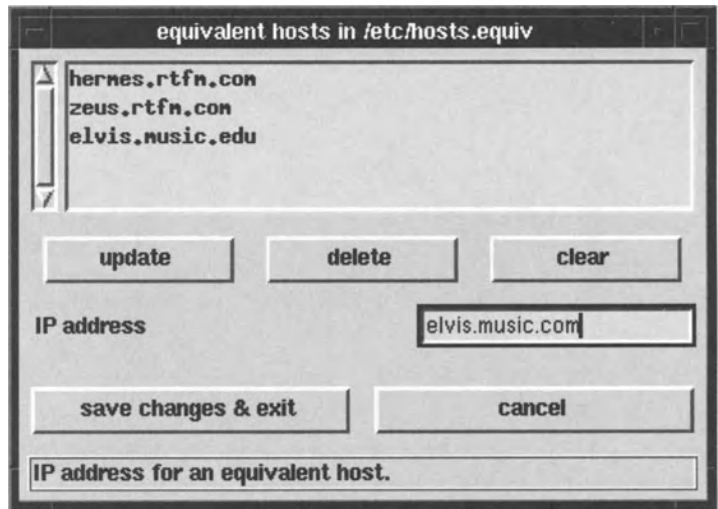
remote access

The daemon `lpd` also uses the file `hosts.equiv`, but also searches for permissions in the file `/etc/hosts.lpd`, which has the same format as the file `hosts.equiv`. If a client only needs access to a printer queue, then for security reasons the name of the client should be entered only in the file `/etc/hosts.lpd`.

The following demonstrates the format of the files `hosts.equiv`, `hosts.lpd` and `.rhosts`.

```
#
# Valid hosts and users
#
linux1.rz.fh-heilbronn.de
linux2.rz.fh-heilbronn.de      strobelt
sun1.rz.fh-heilbronn.de       arnold
```

The files `/etc/hosts.equiv` and `/etc/hosts.lpd` can also be maintained with `xadmin`. The operation is only explained for the module "equivalent hosts", the module "printing hosts" is operated in the same way.



hosts.equiv in xadmin

hosts.equiv

The list at the top contains the list of hosts in the way they are specified in `/etc/hosts.equiv`. To edit or delete a host, click on it with the left mouse button. The button "delete" will remove the selected host from the list, the button "update" will add the host displayed in the entry field to the list, and the button "clear" will clear the entry field. To leave the module without applying any changes, press the button "cancel". To quit after writing out your changes to `/etc/hosts.equiv`, press the button "cancel".

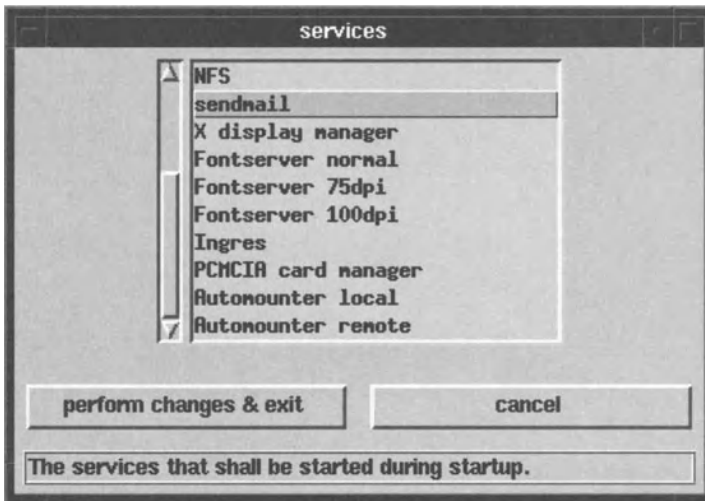
.rhosts

Only the system administrator can modify the files `/etc/hosts.equiv` and `/etc/hosts.lpd`. If an individual user wants to grant permission to another user, an entry in the file `.rhosts` in the granting user's home directory suffices. This approach proves especially practical when a user has different user IDs on multiple machines on the network. Corresponding modification of the `.rhosts` file significantly eases access to the user's own directories throughout the network.

Daemons

`xadmin's module services` allows you to specify the daemons that should be started during the system startup procedure as well

as their configurations. Simply toggle the services in the list by clicking on them with your left mouse button. When you are done, press "perform changes & exit" to write out your changes to `/etc/SERVICES`, restart them and exit; or press "cancel" to exit without applying any changes.



the services module

NFS

The Network File System (NFS) requires several daemons. Since NFS is based on Remote Procedure Calls (RPC), the portmap daemon is needed first. It registers the RPC services of a server and returns the TCP/IP port number to the client submitting requests.

portmap daemon

The NFS daemon `rpc.nfsd` replies to the read and write requests from NFS clients. The daemon `rpc.mountd` is responsible for the mounting itself; it manages directories and checks the permissions of a client submitting a mount request.

rpc.nfsd

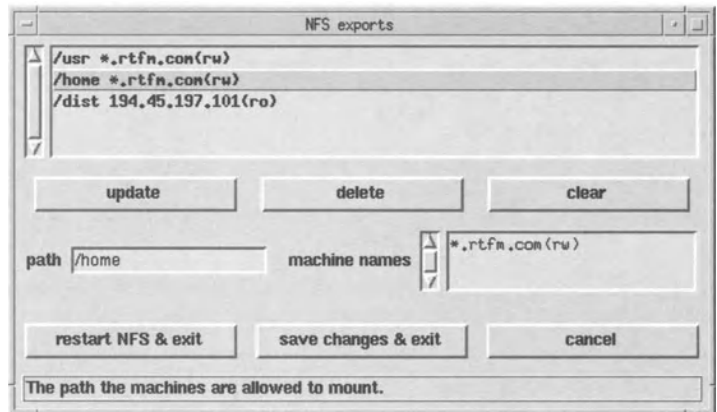
This daemon's most important configuration file is `/etc/exports`, which lists all directories that can be mounted by other machines via NFS along with their permissions.

Modifications in this file become effective only after both `rpc.nfsd` and `rpc.mountd` are restarted.

The following is an example of this file:

```
#
# Exported directories
#
/                linux1(rw)
/home            141.7.1.49(rw)
/home/prog       riscl(rw)
```

This file can be edited with `xadmin`, which also enables the user to restart the respective daemons. The module "exports" is located in the sub menu "networking".



`xadmin`'s exports module

The list at the top contains the current settings in the way they will be written to `/etc/exports`. To edit a line, click on it with the left mouse button and edit the fields "mount point" and "machine names". To apply the changes to the list at the top, press the button "update". To delete a line, select it and press "delete". The button "clear" will initialize the fields "mount point" and "machine names". Entering a new mount point and pressing "update" will add it to the list. To perform the changes to the file `/etc/exports` and to restart the NFS daemons, you need to press the button "restart NFS & exit". The button "save changes & exit" will write out the file `/etc/exports` without

restarting the daemons. By pressing the button "cancel", you exit the module leaving your system unchanged.

Configuration of the kernel

Another step after the base installation of the operating system is the configuration and compilation of the kernel. Although in many cases this can be omitted, in order to achieve optimal fine-tuning to the available hardware, we absolutely recommend compilation. This allows you to dispense with drivers that are not needed, to add new drivers, and to modify settings of drivers. This customizing reduces the memory requirements of the kernel and accelerates booting.

fine tuning

Normally the source code of the kernel is located in the directory `/usr/src/linux`. This directory also contains a configuration script that is invoked by the makefile and significantly simplifies the customizing of the kernel. This script is activated with `make config` and permits the setting of the following options:

RAM requirements

`/usr/src/linux`

make config

- Supported file systems
- TCP/IP support
- Use of coprocessor emulator
- Optimization for 80486 processors
- SCSI support
- Drivers for SCSI and network boards
- Parameters for specialized interface boards
- Settings for sound boards

If the kernel fails to automatically recognize certain parameters such as the I/O address or the interrupt number of a board and these parameters have to be specified, then it sometimes does not make sense to change these in the source code of the respective driver. Instead, the options should be passed to the kernel during booting. The parameters can be entered manually during booting with the boot manager. For example, a computer equipped with two identical Ethernet boards to work as a router or gateway requires the data for these boards because the kernel normally looks for only one network board.

IO address, interrupt

append

Compilation of the kernel

	Once all necessary specifications have been made, the dependencies among the individual part of the source code have to be determined anew. This process is started with the invocation of
<code>make dep</code>	<code>make dep</code> . This requires being in the main directory of the kernel source code (<code>/usr/src/linux</code>). Since old object files could still be present when the configuration is modified, these
<code>make clean</code>	should first be removed with <code>make clean</code> .
	Now the actual compilation process can begin. As during the installation and compilation of most other programs, the makefile plays a central role here: It contains the dependencies between the individual source code files and helps to coordinate various scripts for configuration and installation of the kernel.
compressed kernel	By default on PCs a compressed (packed) kernel is generated. An integrated routine decompresses (unpacks) the kernel during booting. This feature of the Linux kernel makes it possible to put a minimal but complete Linux system that supports all hardware (network, streamer, SCSI devices, CD-ROM) onto a single boot disk. The time overhead in decompression is insignificant.
<code>make options</code>	The following are the most important variants in the invocation of the <code>make</code> command.
	<ul style="list-style-type: none"> • <code>make dep</code> - Determines anew the dependencies among source code files, which should be done after any modification in the configuration of the kernel. • <code>make clean</code> - Deletes all object files. On the next compilation, all source code files are compiled from scratch. • <code>make</code> - The kernel is only compiled. No kernel image is created. • <code>make zImage</code> - Compiles the kernel and creates a compressed, bootable kernel image file in the directory <code>/usr/src/linux/arch/i386/boot</code>. • <code>make zliilo</code> - Creates a new kernel image as above and copies it to <code>/vmlinuz</code>. If an old file of the same name is present, it is backed up as <code>/vmlinuz.old</code>. Then the installation script of LILO is invoked (if LILO is installed). The Linux Universe boot manager does not need to be

reinstalled after a change of the kernel unless you also change its name. See page 27 for a description of the Linux Universe boot manager.

- **make disk** - On completion of the compilation, the image file is written directly to the disk currently in drive A. Before this command is invoked, an empty, formatted diskette should be placed in drive A.
- **make modules** - Generates all loadable modules and stores them in the directory `/usr/src/linux/modules`.
- **make modules_install** - Installs the compiled loadable modules in `/lib/modules/<version>`. Which modules are loaded can be specified with `xadmin`.

Configuration with rdev

Various kernel parameters can be configured in the image file after compilation. This includes the device of the root partition and the size of the ramdisk. The program that supports such configuration is `rdev`. Invoking it with the option `-help` displays a list of all options. The most important invocations of this program are the following:

image file

`rdev`

- **rdev -help**
Displays an overview of all possible options.
- **rdev <image file> <root device>**
Example: `rdev /vmlinuz /dev/hda1`
This invocation defines the device from which the root file system is to be mounted during booting of the specified kernel image file. If the specified image file is loaded by the Linux Universe boot manager, then this setting is usually overwritten by the boot manager configuration. Nevertheless this configuration in the kernel is important if the image file is written directly to a diskette with the command `dd`.
- **rdev <ImageFile>**
Example: `rdev /dev/fd0`
Displays the device that is configured as the device of the root file system in the image file.

- **rdev -R <ImageFile> <Flag>**

Example: `rdev -R /vmlinuz 1`

This determines whether the kernel mounts the root partition for reading and writing or as read-only. Many Linux distributions assume that the kernel will mount the root partition as read-only so that the file systems can be checked with `fsck` on booting. After this check, the file systems are mounted for read/write access. If the kernel does not mount the root partition as read-only for booting, then the file system check proves impossible.

Flag 1 indicates read-only mode and Flag 0 read/write.

- **rdev -r <ImageFile> <RamdiskSize>**

Example: `rdev -r /vmlinuz 1440`

This option specifies the size of the ramdisk in the kernel. It is used primarily for boot diskettes. Here the device of the root partition is set to `/dev/fd0` and the size of the ramdisk to 1440, the size of a 3½ inch diskette. During booting the kernel loads the contents of the boot disk into the ramdisk and mounts this as the root file system. The disk drive is then available for other diskettes.

- **rdev -v <ImageFile> <VideoMode>**

Example: `rdev /vmlinuz -1`

Sets the video mode, with which the kernel initializes the graphic board. -1 is normal resolution; -3 prompts for the desired resolution on startup of the kernel.

6.3 X11 configuration

X server	During X11 configuration the X server is set up for the graphic adapter and the monitor by modification of the central
XF86Config	configuration file <code>XF86Config</code> , which is located in the directory <code>/etc</code> or <code>/usr/lib/X11</code> . Normally the configuration of the X server occurs with the installation of the system. However, during installation only known graphic adapters and monitors can be set up and the user does not directly manipulate the configuration file. Therefore we explain this file itself.

The configuration file

The file `XF86Config` consists of several sections. Here we list these sections and describe the most important ones in detail in the subsequent subsections.

`XF86Config`

- **Files** - This section defines the paths that the X server requires: the path to the RGB color table and the font directories.
- **serverFlags** - This section specifies the server's general flags, including defining whether the server can be ended with `<Ctrl-Alt Backspace>` and how the server should react to UNIX signals.
- **Keyboard** - Defines the connected keyboard and the functions of special modifier keys.
- **Pointer** - For adaptation of the mouse driver, this section lists the kind of mouse and the port it uses.
- **monitor** - Specifies limits and timing data for monitors.
- **device** - This section describes graphic adapters.
- **Screen** - Here monitor definitions and graphic adapters are assigned to X servers.

The section Files

The settings for the RGB table and the font paths have usually already been done correctly. The font paths can be specified in individual lines or in a combined line separated by commas. Font servers are specified in the form `transport/hostname:portNumber`, for example `tcp/zeus:7100`. The following example shows a Files section:

RGB table, font paths

```
Section "Files"
RgbPath      "/usr/X11R6/lib/X11/rgb"
# Examples of font server entries:
#  FontPath   "tcp/127.0.0.1:7100"
#  FontPath   "tcp/font.server.de:7100"

FontPath     "/usr/lib/X11/fonts/misc/"
FontPath     "/usr/lib/X11/fonts/Type1/"
FontPath     "/usr/lib/X11/fonts/Speedo/"
FontPath     "/usr/lib/X11/fonts/75dpi/"
FontPath     "/usr/lib/X11/fonts/100dpi/"

EndSection
```

The section serverFlags

NoTrapSignal
DontZap

Here there are only two options. NoTrapSignal is interesting only for debugging purposes, and DontZap disables the possibility to shut down a server with <Ctrl-Alt-Backspace>:

```
Section "serverFlags"
# If the following option is activated, the X server dumps
# a Core file if it receives a signal.
#   NoTrapSignals

# On activation of the following option the sequence
# <Ctrl><Alt><BS> to terminate the server is suppressed.

DontZap
EndSection
```

The section Keyboard

modifier key
modeShift

In this section the keyboard is configured. The choice of protocol should normally be standard. When using a language-specific extension of ASCII, also note that the right <Alt> modifier key (<AltGr>) must be redefined as modeShift in order to easily access certain characters such as "@" and "!" that are normally not part of such extensions. Many example files assume an American keyboard layout and use the <AltGr> key as Compose.

```

Section "Keyboard"
Protocol      "Standard"
# Delay and repeat rates for autorepeat
AutoRepeat   500 5
# Numlock is to be handled by the server
serverNumLock
# Which LEDs can be influenced by the user (e.g. with xset)

#   Xleds      1 2 3
# Function of modifier keys
#   LeftAlt    Meta
#   RightAlt   ModeShift
#   RightCtl   Compose
#   ScrollLock ModeLock

# Switching consoles with SysReq (normally not used in Linux)
#   VTSysReq

# Command to be invoked after opening the virtual terminal

#   VTInit     "command"
EndSection

```

The section Pointer

For the mouse driver it normally suffices to specify the type of mouse driver mouse and the port used. Usually at installation a link is created for the port as `/dev/mouse`, which can be used here.

```

Section "Pointer"
# One of the following mouse protocols must be selected here.
# This might not reflect the name of the mouse.
# Most generic mice and many Logitech mice use the
# Microsoft protocol.
#   Protocol   "Microsoft"
# All normal bus mice:
#   Protocol   "BusMouse"
# Many new serial Logitech mice use the following
# (also see ChordMiddle below):
#   Protocol   "Mouseman"
# Older Logitech mice:
#   Protocol   "Logitech"
# Other mice:
#   Protocol   "MouseSystems"
#   Protocol   "MMSeries"
#   Protocol   "PS/2"
#   Protocol   "MMHitTab"

# The following should be used under Linux:
#   Protocol   "Xqueue"
#   Protocol   "OSMouse"

Protocol      "Microsoft"

```

```

# Mouse port:
device "/dev/mouse"
# Baud rate and sample rate (needed by some Logitech mice)
#   BaudRate 9600
#   SampleRate 150

# Emulation of a 3-button mouse, die 3rd mouse button
# is simulated by simultaneously pressing the left and right
# mouse buttons.
#   Emulate3Buttons

# ChordMiddle is an option for some 3-button Logitech
# and Mouseman mice.
#   ChordMiddle

# Sets the DTR line of the serial mouse port to 0.
# This option is required for some MouseSystems mice.
# Some mice might also need the ClearRTS option,
# which sets RTS to 0.
#   ClearDTR
#   ClearRTS

EndSection

```

The section monitor

monitor specs
sweep frequency

This section has multiple functions. It sets the ranges and timing data of the monitors. Each monitor is assigned an identifier that is later used to reference it. The data to be specified are the maximum horizontal and vertical sweep frequencies as well as the bandwidth. These data can be found in the technical documentation of the monitors. If nothing else is specified, the bandwidth is assumed in megahertz, the range for horizontal synchronization in kHz, and the vertical refresh frequency in Hz. The specification of these values in the configuration serves to protect the monitor. On startup the server checks whether a specified mode exceeds the range of the monitor and discards the mode if this is the case. After the technical specifications of the monitor, various video modes are listed that are especially adapted to the respective monitor. Their definition is discussed in detail in a subsequent section.

```

Section "monitor"
Identifier "VESA Generic monitor"
VendorName "Unknown"
    ModelName "Unknown"
    BandWidth 300
    HorizSync 23-38
VertRefresh 50-60
# 640x480@60Hz Non-Interlaced mode
# Horizontal Sync = 31.5kHz
ModeLine "640x480" 25 640 664 760 800 480 491 493 525
# 640x480@64Hz Non-Interlaced mode
# Horizontal Sync = 33.7kHz
#ModeLine "640x480" 28 640 664 704 832 480 489 492 525

```

```
# VESA 640x480@72Hz Non-Interlaced mode
# Horizontal Sync = 37.9kHz
#ModeLine "640x480" 31.5 640 664 704 832 480 489 492 520
# VESA 800x600@56Hz Non-Interlaced mode
# Horizontal Sync = 35.1kHz
#ModeLine "800x600" 36 800 824 896 1024 600 601 603 625
# VESA 800x600@60Hz Non-Interlaced mode
# Horizontal Sync = 37.9kHz
#ModeLine "800x600" 40 800 840 968 1056 600 601 605 628
# VESA 800x600@72Hz Non-Interlaced mode
# Horizontal Sync = 48kHz
#ModeLine "800x600" 50 800 856 976 1040 600 637 643 666
# VESA 1024x768@60Hz Non-Interlaced mode
# Horizontal Sync = 48.4kHz
#ModeLine "1024x768" 65 1024 1032 1176 1344 768 771 777 806
# 1024x768@42.6Hz, Interlaced mode
# Horizontal Sync = 34.8kHz
#ModeLine "1024x768" 44 1024 1040 1216 1264 768 777 785 817 Interlace
# 1024x768@43.5Hz, Interlaced mode (8514/A standard)
# Horizontal Sync = 35.5kHz
#ModeLine "1024x768" 45 1024 1040 1216 1264 768 777 785 817 Interlace
# VESA 1024x768@70Hz Non-Interlaced mode
# Horizontal Sync=56.5kHz
#ModeLine "1024x768" 75 1024 1048 1184 1328 768 771 777 806
# 1024x768@76Hz Non-Interlaced mode
# Horizontal Sync=62.5kHz
#ModeLine "1024x768" 85 1024 1032 1152 1360 768 784 787 823
# 1152x900@60.14Hz, Non-Interlaced mode
# Horizontal Sync=57.4kHz
#ModeLine "1152x900" 85 1152 1192 1384 1480 900 905 923 955
# 1152x900@48.5Hz, Interlaced mode
# Horizontal Sync=45.6kHz
#ModeLine "1152x900" 62 1152 1184 1288 1360 900 898 929 939 Interlace
# 1152x900@48.5Hz, Non-Interlaced mode
# Horizontal Sync=76.1kHz
#ModeLine "1152x900" 110 1152 1284 1416 1536 900 902 905 941
# 1280x1024@44Hz, Interlaced mode
# Horizontal Sync=51kHz
#ModeLine "1280x1024" 80 1280 1296 1512 1568 1024 1025 1037 1165 Interlace
# 1280x1024@61Hz, Non-Interlaced mode
# Horizontal Sync=64.25kHz
#ModeLine "1280x1024" 110 1280 1328 1512 1712 1024 1025 1028 1054
# 1280x1024@70Hz, Non-Interlaced mode
# Horizontal Sync=74.4kHz
#ModeLine "1280x1024" 125 1280 1296 1552 1680 1024 1024 1032 1062
# 1280x1024@74Hz, Non-Interlaced mode
# Horizontal Sync=78.85kHz
#ModeLine "1280x1024" 135 1280 1312 1456 1712 1024 1027 1030 1064
EndSection
Section "monitor"
Identifier "EIZO FlexScan T660"
VendorName "EIZO"
ModelName "FlexScan T660i-T/TCO"
    BandWidth 135.0
    HorizSync 30.0-82.0
VertRefresh 55.0-90.0
#ModeLine "1024x768" 80 1024 1088 1152 1280 768 770 772 778
#ModeLine "1280x1024" 135 1280 1328 1408 1688 1024 1025 1026 1060
#ModeLine "1536x1152" 168 1536 1616 1760 2048 1152 1154 1158 1188
EndSection
```

The section Device

This section specifies the available graphic adapters. It can occur more than once. For many boards neither the chipset nor the clock frequencies need to be specified because the server detects these data on startup. For complex boards it can be necessary, however.

graphic adapter

```

Section "device"
Identifier "Generic VGA 16 Color"
#server "XF86_VGA16"
VendorName "GENERIC"
BoardName "GENERIC"
EndSection
Section "device"
Identifier "Generic SVGA"
#server "XF86_SVGA"
VendorName "GENERIC"
BoardName "GENERIC"
VideoRam 1024
EndSection
Section "device"
Identifier "Generic SVGA, VideoRam limited to 1MB"
#server "XF86_SVGA"
VendorName "GENERIC"
BoardName "GENERIC"
VideoRam 1024
EndSection
Section "device"
Identifier "Sigma Legend ET-4000"
#server "XF86_SVGA"
VendorName "Sigma"
BoardName "Sigma Legend ET-4000"
option "legend"
EndSection
#From: koenig@tat.physik.uni-tuebingen.de (Harald Koenig)
#Date: Sun, 25 Sep 1994 18:55:42 +0100 (MET)
Section "device"
Identifier "Miro 10SD GENDAC"
#server "XF86_S3"
VendorName "MIRO"
BoardName "10SD GENDAC"
#   Clocks 25.255 28.311 31.500 0 40.025 64.982 74.844
#   Clocks 25.255 28.311 31.500 36.093 40.025 64.982 74.844
ClockChip "s3gendac"
RamDac "s3gendac"
EndSection

```

clock rate Sometimes it makes sense to specify the clock rate even if the server can detect it automatically since the detected clock rate is used as an identifier. The definitions of video modes refer to this identifier in their mode lines of the monitor definitions. If fluctuations occur during the detection of the clock rate and a clock were rated at 50 instead of 49.5, then on startup the X server might not identify the frequency used by a video mode and thus terminate with an error message.

automatic detection Furthermore, the detection of clock frequency by the server can cause problems with some hardware. Specifying the clock frequencies in the graphic adapter definition disables automatic detection.

determining clock rate To determine the available clock rates, remove the clocks line from the definition of the board and start the X server with the option `-probeonly`. This returns the detected clock rates and other driver information in text mode and then terminates.

The section Screen

In this section a monitor and a graphic adapter can be assigned for every specialized server (superVGA server, mono server, S3 server, etc.). On startup, the server selects the appropriate screen and thus has the data for the graphic adapter and the monitor.

In addition, possible video modes are listed here that refer to the mode lines of the corresponding monitor section. The key combinations <Ctrl-Alt +> and <Ctrl-Alt -> of the numeric keyboard switch between modes.

switching video modes

```
Section "Screen"
Driver "vga256"
device "Generic SVGA"
monitor "IDEK VisionMaster 17 (1)"
Subsection "Display"
Modes "1280x1024" "1024x768" "800x600" "640x480"
EndSubsection
EndSection
Section "Screen"
Driver "accel"
device "Miro 20SD"
monitor "IDEK VisionMaster 17 (1)"
Subsection "Display"
Modes "1024x768" "800x600" "640x480"
EndSubsection
EndSection
```

Setting the video modes

The most difficult and dangerous part of the configuration is the setting of the video modes since this directly defines the sweep frequencies (timing) of the monitor, and a low-priced monitor without protective circuitry against excessive sweep frequencies can be damaged by incorrect values. To prevent such damage, frequency limits of the monitor should be entered in the configuration file from the start.

danger!

The advantage of this kind of configuration of the video mode is that the available monitor can be used to full advantage. For example, a 14 inch Monitor whose maximum horizontal synchronization frequency (HSF) is too low to display 800 x 600 pixels flicker-free could be operated at a resolution of 800 x 550 with 72 Hz screen refresh rate (RR).

optimization

For each mode the configuration file will contain an entry for the dot clock to be used as well as four values each for horizontal

and vertical timing. The specifications can be in a single line (as a mode line) or distributed in multiple lines. The two definitions in the following example are identical:

```
#           Mode      Clock  horizontal      vertical
ModeLine "800x600" 45   800  840 1030 1184    600  600  606  624

# The same mode in different notation:
Mode "800x600"
  DotClock      45
  HTimings      800 840 1030 1184
  VTimings      600 600 606 624
EndMode
```

flags Optionally, flags can be specified at the end of a mode definition. These flags include `interlace`, `+hsync`, `+vsync` and `csync` and affect the interlace mode and the kind of synchronization.

```
Mode "1024x768i"
  DotClock      45
  HTimings      1024 1048 1208 1264
  VTimings      768 776 784 817
  Flags         "Interlace"
EndMode
```

- video mode values
- The respective meanings of the values for horizontal timing are, respectively:
- sync

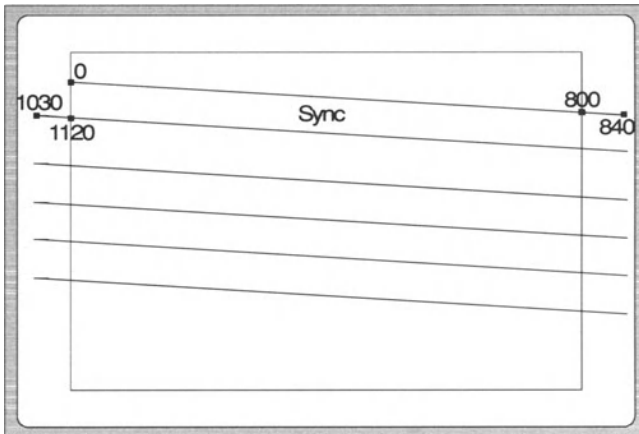
sync duration

guard time

frame
- the maximum number of pixels after which the picture ceases to be displayed
 - the number of dot-clock ticks to the start of the horizontal synchronization pulse (sync), whereby the values are counted ongoing
 - the number of dot-clock ticks until sync ends and the second guard time of the electron beam begins
 - the total number of dot-clock ticks to the end of a cycle (frame)

```
ModeLine "800x5" 45 800 840 1030 1120 540 540 546 558
```

The following example defines a mode with a resolution of 800 x 540 pixels and calls it "800x5".



Schematic representation of picture structure

The horizontal resolution is 800 pixels, and a guard time begins after the end of the visible line. The guard time extends until dot-clock tick 840, when the synchronization pulse begins, lasting 190 dot-clock ticks until 1030, followed by a guard time until 1120, when the next horizontal cycle begins.

After 540 horizontal cycles, or lines, the vertical synchronization begins, lasting six horizontal cycles, and followed by another guard time until 558. Then comes the next picture.

The files `video.tutorial` and `VideoModes.doc` in the directory `/usr/lib/X11/doc` describe in depth the rules for determining individual values for such a video mode. However, it is often simpler to find a corresponding entry in one of the many example files and to modify it. There is also a table for the simple spreadsheet program `sc` that simplifies computations. This table can be found on the ftp server `sunsite.unc.edu` and its mirror servers in the directory `/pub/Linux/X11/install` with the file name `modegen.taz`.

computing values

The limiting factor for simple monitors is usually the maximum horizontal synchronization frequency. This is the frequency with which the electron beam moves from left to right and from scan line to scan line. This frequency is computed by dividing the driving clock rate, specified in MHz, by the largest (right) number of the block for horizontal timing:

horizontal timing

$$f_{horizontal} = \frac{f_{pixel}}{N_{pixel}}$$

In the above example the required horizontal sweep frequency would be 45 MHz / 1120, or approximately 40 kHz. This also happens to be the upper limit for the monitor in our example.

Screen refresh rate

To compute the screen refresh rate (vertical synchronization frequency or vertical timing), divide the horizontal synchronization frequency by the number of scan lines (i.e., horizontal cycles) necessary for a complete screen. This is the rightmost number in the block for vertical timing.

$$f_{vertikal} = \frac{f_{horizontal}}{N_{Zeilen}}$$

In our example we have 40 kHz / 558, or 72 Hz. If 540 lines rather than 600 are to be displayed, then the vertical synchronization frequency falls significantly below 72 Hz, which the user notices as light screen flicker.

screen flicker

improving performance

With a better monitor whose maximum horizontal synchronization frequency is, for example, 60 kHz, and a newer graphic adapter that offers a faster clock, the driving clock frequency could be raised to achieve a higher vertical sweep frequency.

modification

To modify an existing video mode, we highly recommend copying and modifying the mode repeatedly and entering the modified modes in the mode lines of the Screen section with different names. Then start the X server and compare the effects of the modifications by switching modes with <Ctrl-Alt> and the + or - key on the numerical keypad.

danger!

If the monitor no longer synchronizes with a new mode, i.e., it fails to show a stable picture, quickly change modes or end the X server with <Ctrl-Alt-Backspace> to avoid damage to the monitor.

The program `vgaset` provides a valuable aid in adjusting the picture. Started in an `xterm`, it permits interactive manipulation of the picture position. At the touch of a key the borders can be increased or decreased, and the duration of the synchronization signal can be changed. The eight values to be entered for the current settings in the file `XF86config` are constantly displayed.

6.4 Configuration of X applications

Most X clients are provided with an *application defaults* file, which is copied into the library directory of the X11 system (`/usr/lib/X11/app-defaults`). This file contains various base settings that are important for the application, such as size, position and color of the graphical objects or error message texts in the localized language. Every application is assigned a class name by its programmer to which the name of the resource file corresponds. Class names always begin with a capital letter. To change the background color of `xterm` (class name `XTerm`), modifications are made in the file `/usr/lib/X11/app-defaults/XTerm`.

X client

Various environment variables (`XFILESEARCHPATH`, `XAPPLERESDIR`) influence the search path for resource files. `XFILESEARCHPATH` permits the specification of multiple search paths separated by colons and treats certain character combinations specially:

environment variables

search paths

%C	value of customization resource (*.customization)
%L	language, location, codeset
%l	language
%N	class name
%T	filename (app-defaults)

A useful definition of this variable might be the following:

```
XFILESEARCHPATH=/usr/lib/X11/%T:/usr/local/%T/%N:$HOME/%T/%N
```

Thus the following three directories are searched for resource files:

- 1. /usr/lib/X11/app-defaults/<Class>
- 2. /usr/local/app-defaults/<Class>
- 3. <home directory>/app-defaults/<Class>

Another way to configure X11 applications is the command `xrdb`. It loads the transferred resource file into a *property* of the X server (`RESOURCE_MANAGER` or `SCREEN_RESOURCES`). `xrdb` permits a number of parameters:

-all	Operation covers both properties
-screen	Operation covers only the property SCREEN_RESOURCES
-global	Operation covers only the property RESOURCE_MANAGER
-query	Outputs the current contents of a property
-merge <file>	Merges the contents of a file with a property
-edit <file>	Saves the contents of a property to a file
-remove	Removes the complete property
-load <file>	Overwrites a property with the contents of a file

xrdb commands

In addition, each user can create a personal `.Xdefaults` file and customize the default settings.

The following list provides an overview of which files and paths are processed sequentially to determine the current widget attributes (see next section) when an application starts: widget attributes

- **Application-internal:**
 1. Fallback resources
- **Application-specific:**
 1. `/usr/lib/$LANG/app-defaults/<class>`
 2. `/usr/lib/X11/app-defaults/<class>`
- **New search path:**
 1. `$XFILESEARCHPATH`
- **User-specific:**
 1. `$XUSERFILESEARCHPATH`
 2. `$XAPPLRESDIR/$LANG/<class>`
 3. `$XAPPLRESDIR/<class>`
 4. `$HOME/$LANG/<class>`
 5. `$HOME/<class>`
- **Screen-specific:**
 1. `SCREEN_RESOURCES` **property** (xrdp)
- **Display-specific:**
 1. `RESOURCE_MANAGER` **property** (xrdp)
 2. `$HOME/.Xdefaults` **file**
- **Host-specific:**
 1. `$XENVIRONMENT` **variable**
 2. `$HOME/.Xdefaults-<hostname>`
- **Command line:**
 1. Command line option

Widget attributes

class These resource values are represented in ASCII format. For distinction within a resource file, each application is assigned a name (class) by its programmer, which seldom corresponds to the program file (instance). Likewise every widget and widget attribute that can be configured externally possesses an identifier and belongs to a class. For unambiguously referencing a widget, specifying its name alone does not suffice. Analogous to a file system, a path is required that represents a section of the widget-hierarchy.

wildcards To allow simultaneous manipulation of the attributes of multiple widgets, such a path permits wildcards (?, *). Such a resource specification has the following syntax:

```
object.subobject[.subobject...].attributes: value
```

The constituent syntax elements mean the following:

object	class or name of the program
subobject	class or name of the widget
attributes	resource name
value	value
.	separator
*	wildcard = any number of (or no) identifier(s)
?	wildcard = any individual identifier

The first column of the resource file specifies the attribute to be manipulated. This usually corresponds to a widget resource. The programmer can also define new, applications-specific resources. The hierarchy and the names of a program's available resources can be found in the corresponding Manual page.

```
Xterm*background:      gray90
Xterm*ScrollBar:       true
Xterm*Foreground:      white
Xterm*Background:      gray20
Xterm*IconName:        XTerm
Xterm*WaitForMap:      true
Xterm*MarginBell:      false
Xterm*JumpScroll:      true
```

Excerpt from a resource file

Likewise the individual widget attributes can be combined to classes. Taking advantage of class identifiers can make a resource file significantly clearer and smaller. The `xterms` attributes `cursorColor` and `pointerColor` both belong to the class `Foreground`. Therefore the following:

```
Xterm*foreground:       green
Xterm*cursorColor:     green
Xterm*pointerColor:    green
```

can be abbreviated as follows:

```
xterm*Foreground:      green
```

Releases 5 and 6 of the X Window system include an interactive resource manager (`editres`) that permits comfortable manipulation of all resource values of a running program and to save them in an ASCII file if desired. Note that this is possible at run time of a program. In this way the user can immediately see the effects of changes. Unfortunately, the protocol required by `editres` is not yet supported by all widget sets, which naturally limits the usability of the tool. The generated ASCII files can easily be integrated in existing resource files or appended to the file `.Xdefaults`.

Configuration of the Window Manager

The user can not only customize the look and feel of individual applications but also configure most window managers. Since very many Linux users employ `fvwm`, we discuss this window manager in more detail. The `fvwm` parameters are set in the file `system.fvwmrc` in the directory `/usr/lib/X11/fvwm`.

M4-Präprozessor

Alternatively individual users can provide a file named `.fvwmrc` in their home directory.

Using the M4 preprocessor provides additional flexibility. This allows, for example, the linking of additional configuration files or the testing of constraints. The main file (`system.fvwmrc`) thus gains in clarity:

```
#####
#
# system.fvwmrc - fvwm configuration
#
#####
# Paths
ModulePath /usr/lib/X11/fvwm/modules
PixmapPath /usr/lib/X11/pixmaps:/usr/local/lib/pixmaps
IconPath /usr/include/X11/bitmaps/

#####
# External configuration files

include(/usr/lib/X11/fvwm/fvwm.options)

include(/usr/lib/X11/fvwm/fvwm.menus)

include(/usr/lib/X11/fvwm/fvwm.functions)

include(/usr/lib/X11/fvwm/fvwm.bindings)

include(/usr/lib/X11/fvwm/fvwm.styles)

include(/usr/lib/X11/fvwm/fvwm.goodstuff)

include(/usr/lib/X11/fvwm/fvwm.modules)

#####
# Initialization and restart functions

Function "InitFunction"
    Module "I" GoodStuff
    Module "I" FvwmPager 0 1
    Exec "I" exec xterm -sb -sl 400 -geometry +75+390 &
    Exec "I" xsetroot -solid LightSlateGray
EndFunction

Function "RestartFunction"
    Module "I" GoodStuff
    Exec "I" xsetroot -solid LightSlateGray
    Module "I" FvwmPager 0 1
EndFunction
```

system.fvwmrc

In addition to color and font definitions, the file `fvwm.options` contains a number of other options that determine the look and feel.

```
#####
#
# fvwm.options - general options
#

DeskTopSize 2x2
DeskTopScale 32

# Standard colors
StdForeColor      Black
StdBackColor      #d3d3d3

# Window colors
HiForeColor        Black
HiBackColor        #5f9ea0
StickyForeColor    Black
StickyBackColor    #60c0a0

# Menu colors
MenuForeColor      Black
MenuBackColor      grey
MenuStippleColor    SlateGrey

# Fonts
Font               -adobe-helvetica-medium-r--*-12-*-*-*-*-*
WindowFont         -adobe-helvetica-bold-r--*-12-*-*-*-*-*
IconFont           fixed

# Rectangles in which icons are positioned
IconBox 5 -80 -140 -5
IconBox 5 -160 -140 -85
IconBox 5 -240 -140 -165
IconBox 5 -320 -140 -245

# Motif-like look and feel
MM#FunctionHints
MM#HintOverride
MM#DecorHints
MM#Borders
MM#Buttons

# Moves all windows with contents
OpaqueMove 100

# Disable automatic switch of desktop
EdgeScroll 0 0

# Delay on switching the desktop excerpt
EdgeResistance 250 50

NoPPosition

# Forces decoration of transient shell
DecorateTransients
```

fvwm.options

The user can create new menus and assign them to user actions.

```
#####
#
# fvwm.menus - Menu configuration
#

Popup "Shells"
    Title "Shells"
    Exec "Mxterm"      exec mxterm &
    Exec "Color XTerm"  exec color_xterm &
    Exec "Rcvt"         exec rcvt &
EndPopup

Popup "Editors"
    Title "Editors"
    Exec "GNU emacs"    exec emacs &
    Exec "NEdit"        exec nedit &
    Exec "Textedit"     exec textedit &
EndPopup

Popup "Graphics"
    Title "Graphics / Viewer"
    Exec "XPaint"       exec xpaint &
    Exec "XV"           exec xv &
EndPopup

Popup "Modules"
    Title "Modules"
    Module "GoodStuff"   GoodStuff
    Module "Identify"    FvwmIdent
    Module "SaveDesktop" FvwmSave
    Module "Pager"       FvwmPager 0 1
    Module "FvwmWinList" FvwmWinList
    Module "FvwmIconBox" FvwmIconBox
EndPopup

Popup "Window Ops"
    Title "Window Ops"
    Move "%Move Alt+F7"
    Resize "%Size Alt+F8"
    Lower "%Lower Alt+F3"
    Raise "%Raise "
    Stick "(Un)Stick "
    Iconify "(Un)Mi&nimize Alt+F9"
    Maximize "(Un)Ma&ximize Alt+F10"
    Maximize "(Un)Maximize Vertical " 0 100
    Nop ""
    Destroy "%Kill Alt+F4"
    Delete "Delete "
EndPopup

Popup "Window Ops2"
    Move "%Move Alt+F7"
    Resize "%Size Alt+F8"
    Iconify "(Un)Mi&nimize Alt+F9"
    Maximize "(Un)Ma&ximize Alt+F10"
    Lower "%Lower Alt+F3"
    Nop ""
    Destroy "%Kill Alt+F4"
    Delete "Delete "
    Nop ""
    Module "ScrollBar" FvwmScroll 2 2
EndPopup

#####
#
# Main menu

Popup "Programs"
    Title "Programs"
    Exec "Xterm"      exec xterm -sb -sl 400 &
    Popup "Shells"    Shells
    Popup "Editors"    Editors
    Popup "Graphics"   Graphics
    Popup "Modules"    Modules
    Exec "Screen Lock" exec xlock &
    Nop ""
    Restart "Restart Fvwm" fvwm
    Quit "Exit"
EndPopup
```

fvwm.menus

Within the `fvwm` configuration file new functions can be defined, usually being assigned to a keyboard or mouse action.

```
#####
#
# fvwm.functions - function definition
#

Function "Move-or-Raise"
    Move      "Motion"
    Raise     "Motion"
    Raise     "Click"
    RaiseLower "DoubleClick"
EndFunction

Function "maximize_func"
    Maximize  "Motion" 0 100
    Maximize  "Click" 0 80
    Maximize  "DoubleClick" 100 100
EndFunction

Function "window_ops_func"
    PopUp     "Click"      Window Ops2
    PopUp     "Motion"     Window Ops2
EndFunction

Function "Move-or-Lower"
    Move      "Motion"
    Lower     "Motion"
    Lower     "Click"
    RaiseLower "DoubleClick"
EndFunction

Function "Move-or-Iconify"
    Move      "Motion"
    Iconify   "DoubleClick"
EndFunction

Function "Resize-or-Raise"
    Resize    "Motion"
    Raise     "Motion"
    Raise     "Click"
    RaiseLower "DoubleClick"
EndFunction
```

fvwm.functions

The assignment of keyboard or mouse input to the corresponding action occurs in the file `fvwm.bindings`.

```
#####
#
# fvwm.bindings - keyboard and mouse configuration
#
# Structure of a configuration line:
#
#      <key>      <context>  <modifier> <function>
#
#      <key>      (mouse)key
#      <context>   R - root window
#                  W - application window
#                  T - title bar
#                  S - window page
#                  F - window frame
#                  I - icon
#                  A - everything except title bar
#                  0,1,2,... - window control elements
#      <modifier> N - no modifier key
#                  A - alternate
#                  C - control
#                  M - meta
#                  S - shift
#                  mod1-mod5 - X11 modifiers
#      <function> Fvwm-function
#
# Mouse click on root window
Mouse 1 R A PopUp "Programs"
Mouse 2 R A PopUp "Window Ops"
Mouse 3 R A Module "FvwmWinList" FvwmWinList Transient
# Window control elements
Mouse 0 1 A Function "window_ops_func"
Mouse 0 2 A Function "maximize_func"
Mouse 0 4 A Iconify
Mouse 1 F A Function "Resize-or-Raise"
Mouse 1 TS A Function "Move-or-Raise"
# Icon actions
Mouse 1 I A Function "Move-or-Iconify"
Mouse 2 I A Iconify
# Window operations
Mouse 2 FST A Function "window_ops_func"
Mouse 3 TSIF A RaiseLower
# Keyboard shortcuts
Key F1 A M PopUp "Window Ops"
Key F2 A M PopUp "Programs"
Key F3 A M Lower
Key F4 A M Destroy
Key F5 A M CirculateUp
Key F6 A M CirculateDown
Key F7 A M Move
Key F8 A M Resize
Key F9 A M Iconify
Key F10 A M Maximize
```

fvwm.bindings

The file `fvwm.style` specifies the look and feel of individual applications.

```
#####
#
# fvwm.styles - Style configuration
#
Style "" BorderWidth 7, HandleWidth 5
Style "FvwmPager" Sticky, NoTitle
Style "FvwmBanner" StaysOnTop
Style "GoodStuff" Sticky, WindowListSkip, NoTitle
Style "xterm" Icon terminal.xpm
Style "xcalc" Icon rcalc.xpm
Style "xman" Icon xman.xpm
Style "xvgr" Icon graphs.xpm
Style "Mail" Icon sndmail.xpm
Style "emacs*" Icon editor2.xpm
```

fvwm.styles

Each fvwm module has its own configuration possibilities collected in the file fvwm.modules.

```
#####
#
# fvwm.modules - Module configuration
#
##### No Clutter #####
*FvwmNoClutter 3600 Iconify 1
*FvwmNoClutter 86400 Delete
*FvwmNoClutter 172800 Destroy

##### Window Identifier #####
*FvwmIdentBack MidnightBlue
*FvwmIdentFore Yellow
*FvwmIdentFont -adobe-helvetica-medium-r--*-12-*-*-*-*-*

##### FvwmWinList #####
*FvwmWinListBack #d3d3d3
*FvwmWinListFore Black
*FvwmWinListFont -adobe-helvetica-bold-r--*-10-*-*-*-*-*
*FvwmWinListAction Click1 Iconify -1,Focus
*FvwmWinListAction Click2 Iconify
*FvwmWinListAction Click3 Module "FvwmIdent" FvwmIdent *FvwmWinListUseSkipList
*FvwmWinListGeometry +0-1

##### FvwmBackerDesk #####
*FvwmBackerDesk 0 xsetroot -solid steelblue
*FvwmBackerDesk 1 xsetroot -solid midnightblue
*FvwmBackerDesk 2 xsetroot -solid yellow *FvwmScrollBack gray60

##### FvwmIconBox #####
*FvwmIconBoxIconBack #cfcfcf
*FvwmIconBoxIconHiFore black
*FvwmIconBoxIconHiBack #5f9ea0
*FvwmIconBoxBack #cfcfcf
*FvwmIconBoxFore blue
*FvwmIconBoxGeometry 1x5+0+89
*FvwmIconBoxMaxIconSize 64x38
*FvwmIconBoxFont -adobe-helvetica-medium-r--*-12-*-*-*-*-*
*FvwmIconBoxSortIcons
*FvwmIconBoxPadding 4
*FvwmIconBoxLines 10
*FvwmIconBoxPlacement Top Left
#
# mouse bindings
#
*FvwmIconBoxMouse 1 Click RaiseLower
*FvwmIconBoxMouse 1 DoubleClick Iconify
*FvwmIconBoxMouse 2 Click Iconify -1, Focus
*FvwmIconBoxMouse 3 Click Module "FvwmIdent" ndings
#
# Key bindings
#
*FvwmIconBoxKey r RaiseLower
*FvwmIconBoxKey space Iconify
*FvwmIconBoxKey d Close
#
# FvwmIconBox built-in functions
#
*FvwmIconBoxKey n Next
*FvwmIconBoxKey p Prev
*FvwmIconBoxKey h Left
*FvwmIconBoxKey j Down
*FvwmIconBoxKey k Up
*FvwmIconBoxKey l Right
#
# Icon file specifications
#
*FvwmIconBox "*" unknown1.xpm
*FvwmIconBox "Mosaic" www-shape.xpm
*FvwmIconBox "xterm" terminal.xpm
*FvwmIconBox "GoodStuff" toolbox.xpm
*FvwmIconBox "*icon*" daffy.xpm
*FvwmIconBox "*anual*" xman.xpm

##### Pager #####
*FvwmPagerBack #908090
*FvwmPagerFore #484048
*FvwmPagerFont -adobe-helvetica-bold-r--*-10-*-*-*-*-*
*FvwmPagerHighlight #cab3ca
```

```
*FvwmPagerGeometry 0+0
*FvwmPagerLabel 0 Strobel
*FvwmPagerLabel 1 Uhl
*FvwmPagerSmallFont 5x8
```

fvwm.modules

The configuration of the Goodstuff module is located in a separate file named `fvwm.goodstuff`. Goodstuff permits depicting the most important applications in an icon-bar. Clicking on an icon starts the corresponding program.

The Swallow option makes it possible to depict programs in the icon bar, `xload` or `xclock` for example.

```
#####
#
# fvwm.goodstuff - Goodstuff configuration
#
*GoodStuffBack gray60
*GoodStuffGeometry 65x715-1+0
*GoodStuffColumns 1
*GoodStuffFont -adobe-helvetica-medium-r-*-12-*-*-*-*-*
#
#      Name      Icon      Action WindowTitle Command
*GoodStuff "" "" Swallow "xclock" xclock -bg gray60 &
*GoodStuff "" "" Swallow "xload" xload -bg gray60 &
*GoodStuff "" "" Swallow "xbiff" xbiff -bg gray60 &
*GoodStuff XTerm terminal.xpm Exec "xterm" xterm -sb -sl 400 &
*GoodStuff NetScape www.xpm Exec "Netscape" netscape &
*GoodStuff Xman xman.xpm Exec "Manual Page" xman -bothshow -notopbox &
*GoodStuff Mail sndmail.xpm Exec "Mail" xterm -T Mail -e pine &
*GoodStuff Emacs editor.xpm Exec "emacs" emacs &
*GoodStuff Exit lbolt.xpm Quit
```

fvwm.goodstuff

Administration

After completing the installation of the Linux system and the most important configurations, you can permit other users to access the system. In addition, after some time new versions of individual system components will appear and need to be installed to keep up with continuing developments. The associated tasks are collectively termed *system administration*. Since these tasks are very similar on all UNIX systems, we refer you to the standard literature and provide some tips and notes concerning Linux-specific details.

users

updates

system administration

7.1 The administrator

Only the user *root*, the system administrator, can modify the system configuration files. This privilege is protected by corresponding permissions on the files.

root

permissions

Generally the system administrator has read/write permissions on all files. This also means that the administrator can destroy or delete the whole system with erroneous input. For example, if the administrator were to launch the following command to remove all files in a directory with all its subdirectories without asking for confirmation, and do so from the root directory (*/*), then the complete system would be obediently deleted.

```
linux1:/> rm -rf *
```

If such an error were to occur without root privileges, then the permissions of the system files and directories would assure

login precaution

that at worst all the user's own files were removed. This would not influence other users or the overall system.

Thus the system administrator needs to take every precaution and only log in as *root* if a system file actually needs to be modified or a new program is to be installed.

7.2 Shutdown

As with all UNIX systems, a Linux computer must not be simply turned off. Instead the system must be shut down as defined by the command `shutdown`.

cache
i-node table

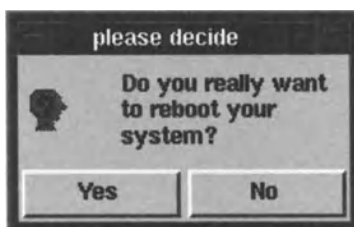
The reason for this restriction is that usually all data that have been written by programs to the hard disk interface have not yet actually been written to the hard disk from the kernel's internal cache. In addition, frequently used information, such as the i-node table or the superblock of the file system, are likewise held in RAM. If the computer is turned off without first running the command `shutdown`, this can produce inconsistencies on the hard disk and data losses.

shutdown

The command `shutdown` assures that all buffers are transferred to the storage media and that all processes terminate properly.

xadmin

The system can also be shut down from `xadmin`. In the sub menu "system" you may choose to reboot or halt your system.



7.3 The directory tree

To help new Linux users and inexperienced system administrators gain some orientation in the system, we will now

explain the most important directories of the Linux Universe system.

The directory `/` is the root of the Linux directory tree. It is thus also called *root* or *root directory*. Except Linux kernel image files that are needed for booting and the most important subdirectories, it should contain no further directories.

root directory

image files

Subdirectories of the root directory

- **/a** and **/amd** - These directories are used internally by the `amd` automounter. automounter
- **/bin** - The most important programs, specifically those that must be present if the directory `/usr` proves inaccessible, reside in this directory. These include the commands `mv`, `cp`, `cat` and `rm`. Unlike `/sbin`, which contains only vital programs for system administration and for booting, `/bin` contains programs intended for all users. All other commands needed in case of an emergency reside in `/usr/bin` (see also `/usr`). programs
- **/dev** - As the name `/dev` indicates, this directory contains device drivers, which are special files that correspond to an input/output driver. devices
- **/dist** - The Linux Universe distribution CD is mounted here. Packages and programs that were not installed on the hard disk have links to files in subdirectories of `/dist`. CD
- **/etc** - The directory `/etc` contains local configuration files, including the files `passwd` and `group`, with user and group information, respectively, and the configuration files for the TCP/IP daemon, such as `services`, `inetd.conf` and `exports`. configuration files
- **/etc/Isode** - The configuration files for the Isode package reside here. Isode
- **/etc/X11** - A number of X11 configuration files, such as `xdm` and the font server, reside in this directory. X11 configuration
- **/etc/amd** - The map files of the automounters reside here.
- **/etc/cnews** - This directory contains the configuration files of the news server `cnews`. news server

- **/etc/crontabs** - Script files that are called from cron reside in this directory.
- defaults • **/etc/default** - Programs for system administration can store default settings in this directory.
- WWW server • **/etc/httpd** - The configuration files for the http server (for World-Wide Web) reside in this directory.
- ingres • **/etc/ingres** - A number of files in the `ingres` database are accessed via symbolic links in this directory.
- rc scripts • **/etc/init.d** - Actual `rc` scripts used in booting and shutdown reside here. These scripts are invoked via symbolic links from the directories `/etc/rc0.d` to `/etc/rc6.d`.
- news server • **/etc/inn** - The configuration files of the news server `inn` reside in this directory.
- WWW client • **/etc/lynx** - The configuration files of the WWW client `lynx` reside here.
- mgetty, sendfax • **/etc/mgetty+sendfax** - Files for the `mgetty` and `sendfax` packages reside in this directory.
- communications • **/etc/minicom** - The communications program `minicom` stores its global settings here.
- news readers • **/etc/news** - This directory is used by the `nntpd` daemon and some news readers.
- window manager • **/etc/openwin** - The menus of the `olwm` and `olvwm` window managers are stored here.
- PPP config • **/etc/ppp** - PPP configuration files reside in this directory.
- run level • **/etc/rc0.d** to **/etc/rc6.d** - The scripts in these directories are executed by `init` on changes of the run level. Normally these files do not reside directly in the directories but are linked to files in `/etc/init.d`.
- new user • **/etc/skel** - On creation of a new user with `xadmin`, the files in this directory are copied automatically to the home directory of the new user. The directory normally contains examples of user-specific configuration files.
- booting • **/etc/rc.d** - The scripts invoked by `init` on booting the system usually reside in this directory. Alternatively they can reside directly in the `/etc` directory.
- **/ftp** - This directory is used by the ftp server daemon. Users logging in to the server with the name `ftp` or `anonymous` can access subdirectories of `/ftp`.

- **/home** - For every user except `root`, a home directory is set up in this directory. The respective subdirectory contains user-specific configuration files. Beyond the user's personal files, no other programs should be installed here.
home directory
- Since this directory usually resides on a separate partition, it is not advisable to locate the home directory of `root` here as well. If this file system cannot be mounted due to some error, then even the administrator might be unable to log in to the system to correct the error.
root
- **/lib** - The images of the most important shared libraries of the system are located in the directory `/lib`. These are the parts of the shared libraries that contain the actual routines and that are loaded with the launching of a program that uses them. The other parts, called the *stubs*, are stored in the directory `/usr/lib`; they are linked to the programs and contain only references to the actual routines. Shared libraries that are not absolutely needed for booting and administration, such as the libraries of the X Window System, should reside in another directory under `/usr`; in the case of X11 this would be `/usr/lib/X11`. `/lib` contains only symbolic links to these libraries.
libraries
- **/linux** - This directory should contain only a symbolic link to the directory with the source code of the Linux kernel.
kernel
- **/local** - A symbolic link from `/usr/local` references this directory. Install local programs here that are not included on the Linux Universe CD.
CD programs
- **/lost+found** - This directory is automatically created on creation of a file system of type `ext2` and is used by utilities such as `fsck`.
file system
- **/mnt**, **/mnt1** to **/mnt3** - These directories should be empty and are frequently used to temporarily mount diskettes or file systems on other computers via NFS.
mount
- **/proc** - This is where the Proc file system is usually mounted. The Proc file system is a special file system in which information on the kernel and processes is represented as subdirectories and files. These files can usually be read as text and thus readily provide information on the processes.
kernel info

admin's home directory

system programs

iBCS2

tftp

temporary files

log and spool files

user programs,
libraries, Manual pages
and configuration

root file system

/dist file system

- **/root** - Although this directory is optional, it is created by most Linux distributions. This is the home directory of the superuser (root). Normally this directory does not reside on the same partition as the home directories of normal users.
- **/sbin** - This directory contains only the most important programs and commands needed for booting the system and for rudimentary system administration. These include `getty`, `init`, `update`, `fdisk`, `fsck`, `ifconfig` and `ping`. Programs that are needed by users other than root reside in `/bin` or in `/usr/bin` if they are not absolutely needed in case of emergency.
- **/shlib** - Shared libraries for the iBCS2 emulation reside here. The directory is a symbolic link to `/usr/lib/ibcs/sci-libs`.
- **/tftboot** - If the `tftpd` daemon is used, then access per `tftp` to this directory can be restricted.
- **/tmp** - This directory is a symbolic link to `/var/tmp`.
- **/var** - In this directory reside all files that are often written to and whose size continually changes.
- **/usr** - This directory contains almost all other important subdirectories that are not directly necessary for booting the system. The separation of machine-specific configuration, essential programs for system administration, and log and spool files from the programs in `/usr` makes it possible to use `/usr` from the CD or for multiple hosts from a common NFS server. Here the `/usr` directory must be mounted as write-protected.

The most important programs for system administration and the required libraries still need to reside in the root file system so that, if a system error makes the CD or the NFS server inaccessible, the error can be corrected. The root file system should be as small as possible to allow shared use of as many programs and storage space as possible.

The Linux Universe CD is mounted in `/dist`, and `/usr` originally contains only symbolic links to directories under `/dist`. The installation program and the CD cache mechanism copy programs from the CD to the correct

location in `/usr` and thus replace the corresponding symbolic links.

Directories under `/usr`

- **`/usr/X11R5`** - This is a link to the directory `/usr/X386`, in which the files of Release 5 of the X Window System are installed. X11R5
- **`/usr/X11R6`** - This is the actual directory of the X Window System starting at Release 6. The directories `/usr/lib/X11` and `/usr/bin/X11` are links to this directory tree. X11R6
- **`/usr/X386`** - This is where the directory tree begins for older versions of X11 packages. Starting with Release 6 only the directory `/usr/X11R6` is used. X11R5
- **`/usr/adainclude`** - All the include files of the Ada compiler reside here. Ada
- **`/usr/adm`** - This is a link to `/var/adm`.
- **`/usr/bin`** - Most system programs and UNIX commands for users reside here along with those for the administrator that are not absolutely necessary in the event that `/usr` proves unmountable. programs and commands
- **`/usr/bin/X11`** - X11 programs are normally installed in this directory. However, this is usually only a symbolic link to `/usr/X386/bin` or `/usr/X11R6/bin` for newer X11 releases. This directory should be contained in the path. X11
- **`/usr/dll`** - This directory is used for DLL tools, with which shared libraries can be created in old formats. DLL
- **`/usr/doc`** - Documentation that is not available as a Manual page or in Info format resides in this directory. documentation
- **`/usr/g++-include`** - Header files for the GNU C++ compiler reside here. header files
- **`/usr/html`** - On-line documentation in html format resides in this directory. It can be read with any WWW client. WWW
- **`/usr/i486-linuxelf`** - This directory is used by the GNU C compiler. GNU C
- **`/usr/i486-unknown-linux`** - This directory is also used by the GNU C compiler.

- **/usr/include** - Include files of the C library reside here. This directory contains the subdirectories `sys` and `linux`, whereby `linux` is a link to a subdirectory of the kernel source code.
- GNU Info files • **/usr/info** - This directory is used for the GNU Info system. The files in this directory can be viewed in info mode in the Emacs editor or with programs like `tkinfo` and provide the primary documentation of GNU programs.
- libraries • **/usr/lib** - The static libraries for various programming languages and the stubs for the shared libraries reside here. In addition, this directory contains multiple subdirectories that usually contain help and configuration files of other programs.
- **/usr/lib/X11** - Here we find the fonts, color tables and other files of the X Window system. This directory is usually a link to `/usr/X11R6/lib/X11`.
- X11 configuration • **/usr/local** - This directory should encompass all programs that are installed, but were not included in the installation package. It contains a complete subdirectory tree consisting of `bin`, `lib`, etc, `include` and `man` directories. As a rule `/usr/local/bin` should be part of the path for programs (`PATH`) and `/usr/local/man` for Manual pages (`MANPATH`). In the Linux Universe Distribution this is a link to the directory `/local`.
- additional installations
- Manual pages • **/usr/man** - The Manual pages reside in subdirectories of `/usr/man`.
- **/usr/openwin** - The subdirectories of `/usr/openwin` contain programs and data of the Sun XView package. The libraries and configuration files usually reside in the directory `/usr/openwin/lib`. The clients of these files are the definition files of the menus for the OpenLook window managers (`olwm` and `olvwm`). The file names all begin with `openwin-menu`. The window manager itself and the other programs reside in `/usr/openwin/bin`.
- OpenWindows / XView
- program packages • **/usr/pkg** - To enable separate installation of individual program packages on the hard disk, these are written to subdirectories under `/usr/pkg`. From there the packages can be installed on the hard disk with `xadmin`.

- **/usr/sbin** - Like the directory **/sbin**, this directory primarily contains programs for system administration, but only ones that are not essential for booting. In addition, the network daemons reside here.
- **/usr/spool** - This is a link to the directory **/var/spool**.
- **/usr/src** - The subdirectories of **/usr/src** contain the source codes of system programs. The most important of these subdirectories is **/usr/src/linux**, where the source code of the Linux kernel resides. source code
- **/usr/tmp** - This is only a link to the directory **/var/tmp**.

Important subdirectories of /var

- **/var/adm** - Log files of the **syslog** daemon and other systems are stored here. log files
- **/var/tmp** - This directory is used by many programs for temporary files. Their access might be via the symbolic link **/tmp**. All users can read and write to this directory. As a rule, files in this directory can be deleted when no application processes are running. No user other than the administrator should be logged in when this deletion is carried out. temporary files
- **/var/run** - Files that are only important during the actual execution of the system reside here. This includes files with the process numbers of various daemons. system files
- **/var/spool** - The queues of the mail and printer systems reside here. queues

7.4 Users and groups

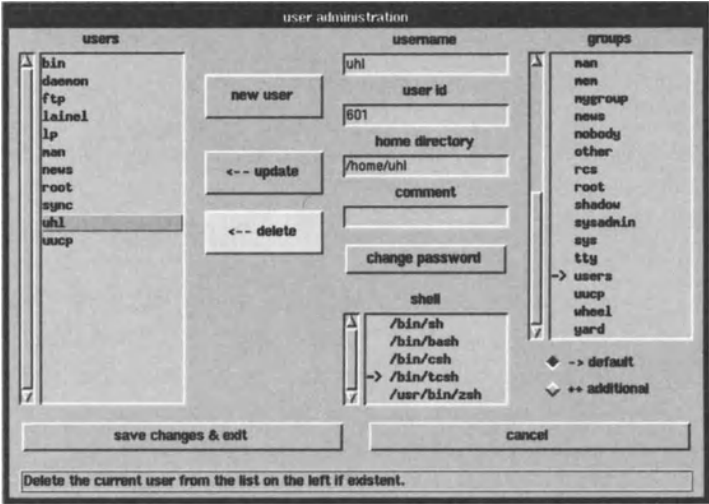
Every user possesses a unique user ID and belongs to one or more groups. This information is stored in the files **/etc/passwd** and **/etc/group**. /etc

To add a new user, you should not modify these files with a text editor. Instead, use the module "users" of the program **xadmin**, which enables you to add and remove users as well as to set users' IDs, home directories, comments, passwords, shells and groups. xadmin

This program manipulates the files **/etc/shadow** (if

xadmin layout

existent), /etc/passwd and /etc/group. If the file /etc/shadow exists, shadowing of passwords is supported. Besides the common elements at the bottom, the xadmin module for user administration displays a list of all users at the left and the settings of the selected user at the right. In between there are three buttons: "new user", "update" and "delete". Be aware that whatever you do, nothing will actually affect your system before you press the button "save changes & exit".



user administration with xadmin

Adding a new user

In xadmin's module "users", click on the button "new user". This will find the first unused user ID and display it in the entry "user ID". A shell and a default group will be preselected for the new user, the password will be empty by default. Enter the user name in the field "username". Optionally, change the home directory by editing the entry "home directory" and add any additional information in the entry "comment". (Changing the

password, shell and groups is described below.) To add the new user to the list on the left, press the button "update".

To select a user, click on the respective user in the list on the left. This will display the corresponding information on the right.

Changing passwords

In the module "users" of `xadmin`, press the button "change password". This will raise a new window with two entries for the password with all characters displayed as asterisks. The second entry assures that you do not enter a misspelled password: If you press "ok" and the passwords don't match, you will be asked to enter them once more. If they do match, the password of the current user will be updated internally. Be aware that the password will not actually be changed until you press "<-- update" to confirm your changes for the current user and "save changes & exit" to write your changes to disk.

change password

Selecting a new shell

In `xadmin`'s module "users", click on the desired shell. To apply the change to the list on the left, press the button "<-- update" to confirm your changes for the current user and "save changes & exit" to actually write your changes to disk.

Selecting default and additional groups

To change a user's default group in `xadmin`'s module `users`, make sure the radio button "-> default" is selected below the list of groups. Then click on the desired group to select it as the new default group. Only one group can be selected as the default group. To remove or add the current user to additional groups, make sure the radio button "++ additional" is selected below the group list. You may now toggle any group except for the selected default group to add it to list of the user's additional groups or remove it. To apply the changes for the current user to the list on the left, press the button "<-- update". Changes

default group

additional group

apply only after you press "save changes & exit" to actually write your changes to disk.

Deleting a user

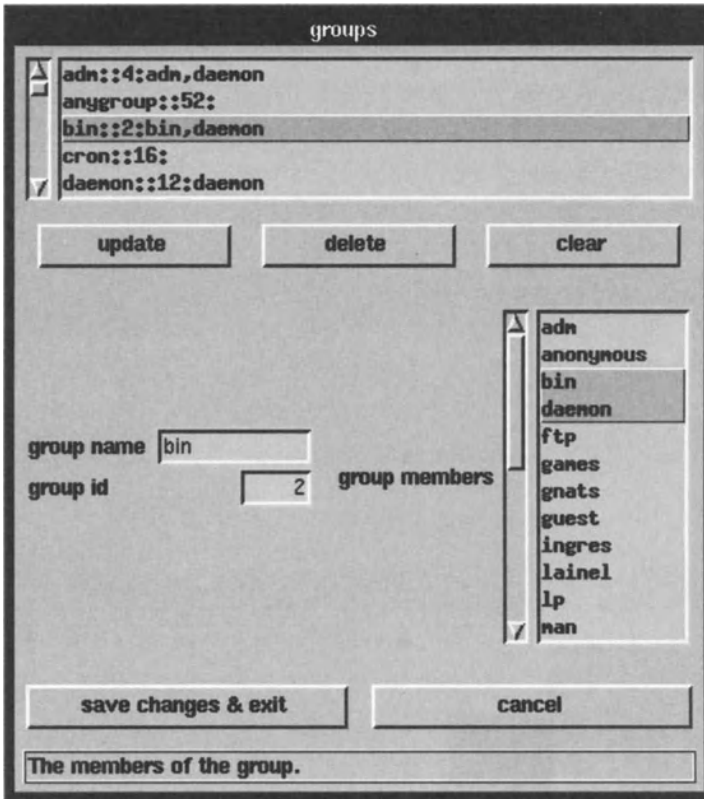
In the module "users" of `xadmin`, select the user and press the delete user button "<-- delete".

Applying your changes

Pressing the button "save changes & exit" will apply the changes to users that you confirmed with the buttons "<-- update" to the configuration files and remove users that were deleted with the button "<-- delete". For new users, the home directory will be created and `/etc/skel/` will be copied into it. Furthermore, the script `/etc/xadmin/xadduser` will be executed with the parameters `username`, `userid`, `userhome` and `usershell`. Editing or replacing this script enables you to add any special actions when a user is added. By default, the script does not perform any further actions. Similarly, when a user is deleted, the script `/etc/xadmin/xdeluser` is executed with the parameters `username`, `userid` and `usergroup`.

Creating new groups

The module "groups" of `xadmin` is used to add and remove groups, and it can be used to add users to or remove users from a specific group. The list at the top contains the group settings of your system as they will be written to `/etc/group` when you press the button "save changes & exit".



/etc/group in xadmin

To edit a group in the list, click on it. You may now edit the entries "group name" and "group id". All users who have the current group as an additional group are selected in the listbox at the right. Click on a user name to toggle the selection. (A user's default group is specified in the file `/etc/passwd`, which can be edited using the module "users".) The button "update" will change the settings for the group in the list or add the group if it is not in the list yet. The button "delete" will remove the selected group from the list. Pressing the button "clear" will display the first unused group ID above 99 in the entry "group ID", clear the entry "group name" and unselect all users in the listbox "users". The setting of group passwords is not supported currently.

edit group

7.5 Shells

/etc/shells

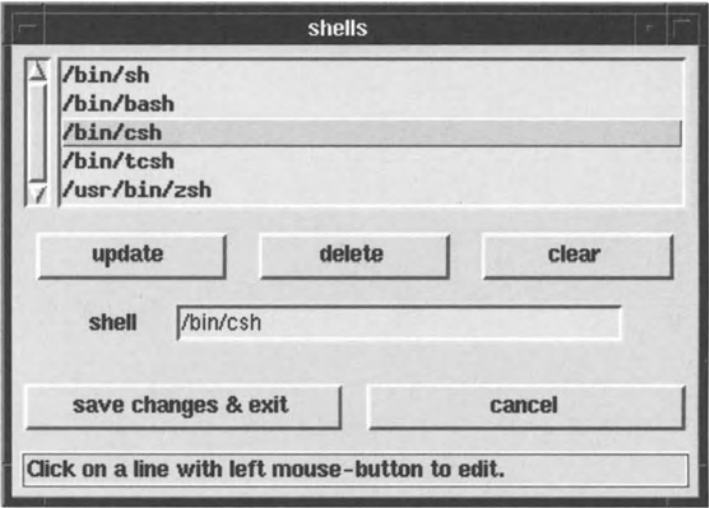
login shell

To allow all users to change their shells independently, each shell and its access path must be listed in the file `/etc/shells`. This is often forgotten when a shell is installed later.

If a certain shell has no entry in the file `/etc/shells`, then users cannot use it as their login shell and there can be problems with certain TCP/IP programs. The following example shows the contents of an `/etc/shells` file.

```
/bin/sh
/bin/bash
/bin/ksh
/bin/tcsh
```

This file can also be modified with `xadmin`.



/etc/shells in xadmin

delete shell
add shell

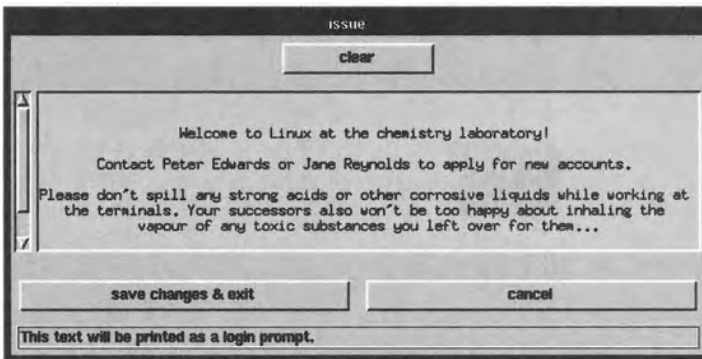
A list of the shells found in the file `/etc/shells` is displayed at the top. To delete a shell, select it by clicking on it and press the button "delete". To add a new shell to the list, enter its path in the entry field "shell" and press the button "update". The button "clear" will clear the entry "shell". The button "save changes &

`exit` will write the new `/etc/shells`, whereas the button `"cancel"` will quit the module without modifying `/etc/shells`.

7.6 User Information

The text displayed before the login prompt resides in the file `/etc/issue`. As a rule this file contains a greeting text including the name of the host and instructions for users. `/etc/issue`

Once the user has logged in, the text in the file `/etc/motd` is displayed. `motd` is an acronym for "message of the day" and should also be used accordingly. Both `/etc/issue` and `/etc/motd` can be modified with `xadmin`. `/etc/motd`



`/etc/issue` in `xadmin`

7.7 Backups

The `tar` command provides a relatively easy way to make backups of important files. This standard UNIX command is available in the Free Software Foundation extended form under Linux. `tar`

For example, to back up all data in the directory `/home/stefan` on diskettes, the `tar` command can be invoked in multivolume mode using the option `M`. When the first diskette is full, `tar` diskettes

requests the next diskette. The following examples demonstrate the invocation of the `tar` command:

```
dirkl:/root# cd /home/stefan
dirkl:/home/stefan# tar cvfM /dev/fd0 *
```

subdirectories Subdirectories are automatically included in the backup. To restore such a backup to the hard disk, the `tar` command must be invoked again with the `M` option; otherwise it will terminate after the first diskette.

multivolume restore

```
dirkl:/root# cd /home/stefan
dirkl:/home/stefan# tar xvfM /dev/fd0
```

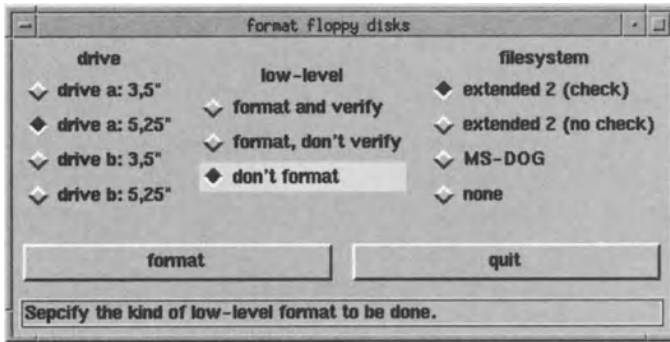
streamer Likewise larger backups can be made on magnetic tape (streamer). Here the streamer's device (`/dev/rmt0`) must be specified rather than `/dev/fd0`.

GNU tar On other UNIX systems that do not use the GNU `tar` command, the option `M` is not available. Therefore for backups that might be restored on another UNIX machine, do not use this option.

7.8 Format and copy diskettes

format The `tar` command as well as the MTools assume the diskettes to be used are already low-level formatted. Likewise the command `mformat` only writes a DOS file system on an already formatted diskette. For the actual low-level formatting, use the command `fdformat` or `xadmin`'s module "format floppy disks". `xadmin` also allows to establish an MS-DOS or extended 2 file system on your diskettes. This can be done by hand using the commands `mkfs` or `mke2fs`.

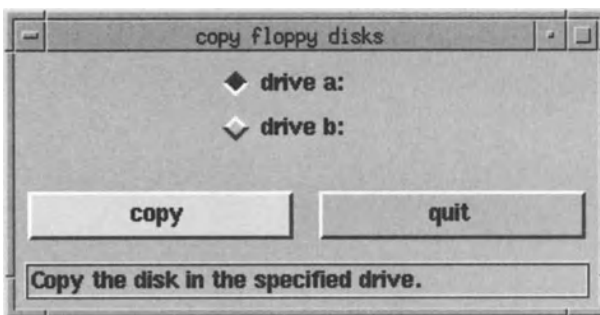
fdformat



formatting floppy disks with xadmin

An easy way to copy diskettes is provided by `xadmin`'s module "copy floppy disks". When you have selected a floppy drive and pressed the the button "copy", it will ask you to insert the source disk and read it in. Then it will ask you to insert the target disk in the same disk and write out the data.

copy floppy disks



copying disks with xadmin

Of course, it is also possible to copy diskettes by hand, for example using `dd`, `mcopy` or `mount` and `cp`.

7.9 File system management

Another area of responsibility of the system administrator is the management of the file systems. Under normal operation this

is limited to checking free storage at regular intervals, monitoring the log files, and from time to time clearing the contents of the /tmp directories.

crash However, if there is a system crash, then a consistency check should absolutely be carried out. Most distributions handle this automatically on booting after every crash. In case of problems, the consistency check must be carried out manually.

booting
file system check There is a special tool for this purpose named `fsck` (file system check). Like `mkfs`, this tool is a front end for other programs that were developed for the respective type of file system. For the currently most popular `ext2` file system, the corresponding tool is `e2fsck`. The system administrator can either invoke `fsck` with the option `-t` and the type of the file system or directly invoke the respective program.

The following example shows the invocation of `e2fsck`:

```
hermes:/root# mount
/dev/sda1 on / type ext2 (defaults)
/proc on /proc type proc (rw)
/dev/sda2 on /usr type ext2 (rw)
/dev/sda5 on /var type ext2 (rw)
/dev/sda6 on /www type ext2 (rw)
/dev/sda7 on /ftp type ext2 (rw)
hermes:/root# umount /ftp
hermes:/root# e2fsck /dev/sda7
fsck.ext2 0.5a, 5-Apr-94 for EXT2 FS 0.5, 94/03/10
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Check reference counts.
Pass 5: Checking group summary information.
/dev/sda7: 9094/212160 files, 750177/845401 blocks
hermes:/root# mount -text2 /dev/sda7 /ftp
hermes:/root#
```

inconsistencies The partition or the file system to be tested is passed as parameter on invocation. Normally only any occurring inconsistencies are displayed. The file system is in order if there are no error messages.

automatic correction Special options also permit automatic error correction. However, the program authors generally recommend interactive repair of a defective file system.

To repair the root file system of the hard disk, we recommend booting with a boot diskette that has its own root file system. This prevents correcting a file system while it is mounted.

7.10 Boot diskettes

It does not take a novice to accidentally delete important files and render the system unbootable. Primarily kernel images that the boot manager loads, shells, and various files in the `/etc` directory are the endangered components here. In such a case there are various possible approaches to repairing the system.

If the system does not start even in single-user mode, you need a boot diskette. If the boot diskette of the Linux Universe Distribution is available, then you can boot the Linux system from it. With the installation program you start a shell, mount the root partition of the hard disk, and reconstruct or copy the missing files. A backup of important files on diskette can be quite helpful.

The Linux Universe boot diskette can also be created under DOS with `rawrite.exe`. This procedure is explained on page 12 of the Installation.

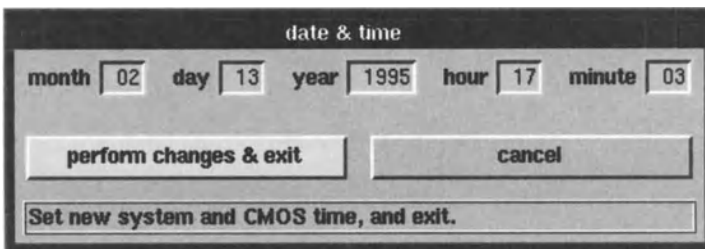
important files

`/etc`

boot disk

7.11 Setting the system time

To adjust your system date and system time, you can use the commands `date` and `clock` or `xadmin`'s module "date & time" in the sub menu "general".



setting the time with xadmin

Edit the values in the entry fields and press the button "perform changes & exit" to set the new system time and exit. To exit without modifying your system time, press the button "cancel".

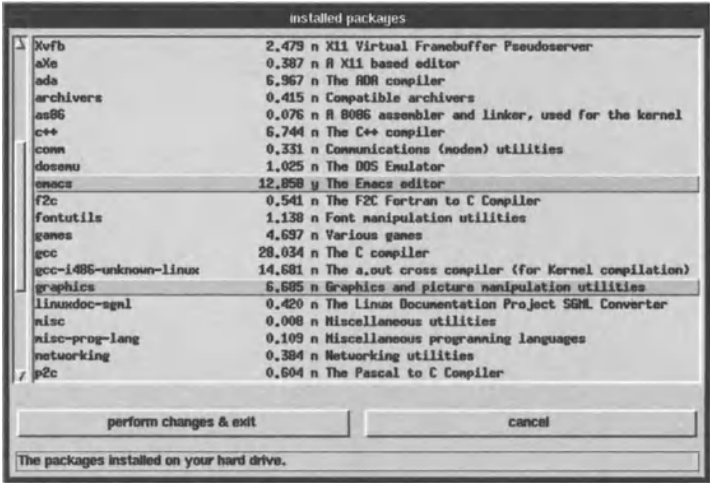
7.12 Linux Universe Packages

variety of packages

Linux Universe offers quite a variety of packages that you may wish to install on your hard drive. A package is a set of files which belong to the same group of programmes, e.g. graphics. Initially, all Linux Universe Packages only exist on the CD-ROM. If you use the programmes in a specific package frequently, it is very useful to install that package on your hard drive. This will make access to the corresponding files much faster, and you don't have to make sure the Linux Universe CD is mounted to /dist when using them.

xadmin

To install new packages or to remove packages previously installed, use xadmin's module "packages" in the submenu "system".



installing Linux Universe packages

This module will display a list of all packages with four columns. The leftmost column is the name of the package, the second column shows the size of the package in megabytes. The third column contains one of the letters **y** (installed), **n** (not installed), **p** (partly installed) or **e** (error). A small description of the package is displayed in the rightmost column. Installed packages are selected in the list. Deselect the packages you want to remove, and select the packages you want to be installed. You

may start the installation and/or removal of the packages by pressing the button "perform changes & exit". Depending on the size of the packages and the speed of your CD-ROM, this process may take quite a while. You are advised to use this time for taking a drink. To leave the module without applying any changes, press the button "cancel".

Reference

apropos terms

Searches the command descriptions in the Manual Pages for the terms passed as parameters and displays the descriptions of the appropriate commands. This is equivalent to the command `man -k`. (Also see `whatis`.)

apropos

```
ar [-]operation [arguments] [position_name]
archive [files]
```

Processes an archive file, which is usually a C compiler library. With this command you can combine any binary files to a library or extract files from a library. Only one operation may be specified, but multiple arguments are permitted.

ar

Available operations

d	deletes the files from the archive
m	moves the files to the archive (with the position depending on additional arguments)
p	lists the files in the archive
q	appends the specified files to the end of the archive
r	replaces the specified files in the archive with the new files
t	lists the contents of an archive (with argument <code>v</code> providing a verbose list output)
x	extracts all files or only those specified from the archive

Available arguments

- a** places the files in the archive after position name
(can be specified with **r** or **m**)
 - b** places the files in the archive before position name
(can be specified with **r** or **m**)
 - c** creates the archive
 - i** see argument **b**
 - o** retains the original file date on extraction
 - s** creates the file table of the archive anew
 - u** replaces only files that have been modified (can be
specified with **r**)
 - v** outputs detailed messages for every operation
-

at [options] time

at

Executes commands at a certain time. The commands are entered at the standard input device and terminated with EOF (<Ctrl-d>). Option **f** permits alternative input from a shell script. A Bourne shell (/bin/sh) is used for execution.

By using the option **q**, individual jobs can be assigned to different queues (a-z, A-Z), where letters later in the alphabet reflect decreasing priority.

The time can be specified in numeric form (HHMM, HH:MM) or with a keyword such as noon, teatime (16:00) or midnight. Alternatively, the time can be specified as a difference such as now +3 hours. Minutes, hours, days, weeks, months and years are permissible units. If the job is to run on a certain day, then the month (Jan, Feb, Mar, ...) and the year (95, 96, ...) are specified additionally.

Available options

- b** equivalent to the command **batch**
- d** removes the specified jobs from the queue (**atrm**)
- f file** executes the commands in *file*
- l** lists the current user's jobs (**atq**)
- m** sends an e-mail to the user when the commands have been completed

-
- q *Q*** assigns a job to a particular queue specified in *Q* (a-z, A-Z)
 - V** returns the version number
-

atq [options]

Displays the jobs that are yet to be executed by the user's at commands.

atq

Available options

- q *Q*** restricts output to the contents of a specific queue *Q*
 - V** displays the version number of the command
 - v** displays a list of jobs that have been executed, but not deleted
-

atrm [options] jobs

Removes the specified at jobs. A job is identified by its job ID, displayed by the at or atq command.

atrm

Available options

- V** displays the version number
-

awk [options] [program] [-v Var=value ...] [files]

awk is a simple interpreter with the combined functionality of grep and sed. It contains its own C-like language. awk is particularly suitable for evaluating ASCII files and for creating scripts for system administration.

awk

Available options

- f *file*** reads the program from the specified file instead of the command line
 - F *c*** sets the delimiting character for fields to *c*
 - v *var=value*** assigns the variable *var* the specified value
-

Reference

basename

basename pathname [suffix]

Clears the path and optionally a specified file extension and outputs the remaining filenames to the standard output device. It is usually used in shell scripts.

bash

bash [options] [arguments]

A command interpreter similar to Bourne shell and Korn shell.

batch

batch [options] [time]

Behaves like the `at` command. However, it only executes the specified commands when the system load is low. (Also see `at`.)

Available options

- f file** executes the commands specified in *file*
 - m** sends an e-mail to the user when the commands have been completed
 - q Q** assigns the job to the queue specified in *Q* (a-z, A-Z)
 - V** displays the version number
-

bc

bc [options] [files]

Interactive program for computation or converting numbers to another base. `bc` has its own language, which supports the definition of new functions, for example.

Available options

- l** makes the functions of the mathematics library available
- s** causes POSIX-compatible behavior
- w** displays warnings that conform to POSIX

Example

```

zeus:/home/uhl> bc
bc 1.02 (Mar3, 92) Copyright © 1991, 1992 Free Software Foundation,
Inc.
This is free software with ABSOLUTLY NO WARRANTY.
For details type 'warranty'.
a=5
b=3
a*b
15
quit
zeus:/home/uhl>

```

cal [options] [[month] year]

Displays a calendar for the current month or a specified month or year. The number of the year must be given in long form (e.g., 1995) and the month as a number (1-12).

cal

Available options

- j displays a Julian calendar (with days numbered sequentially)
- y displays a calendar for the current year

cat [options] [files]

Reads multiple files and outputs them to the standard output device. If no files are specified, the standard input device is read. For this command output is frequently redirected with >.

cat

Available options

- b sequentially numbers all non-empty lines
- e can be specified along with -v and outputs "\$" for end of line (EOL)
- n sequentially numbers all lines
- s replaces a group of blank lines with a single one
- u unbuffered output
- v also outputs control characters and other nonprintable characters
- t can be specified along with -v and outputs "^I" instead of tabs and "^L" instead of page feeds

Example

```
zeus:/home/uhl> cat >file.txt
This is the contents of the file!
<Ctrl-d>
zeus:/home/uhl> cat file.txt
This is the content of the file!
zeus:/home/uhl>
```

cc**cc** [options] files

C compiler (see gcc).

cd**cd** [directory]

Changes the current directory. This command is usually included in the shell. If no directory is specified, the current user's home directory is assumed by default. Specifying "-" returns to the previous directory.

chgrp**chgrp** [options] group files

Changes the group membership of files. This command can be used by the system administrator or by the owner of the specified files. The group can be specified in the form of a numeric group ID or as the name of a group.

Available options

- c** displays the names of the files whose group membership actually changed
- f** suppresses error messages
- R** changes the group membership of files in subdirectories (recursively)
- v** verbosely describes each change

chmod**chmod** [options] permissions files

Changes the permissions of the specified files. This command can be used by the system administrator or the owner of the specified files. The permissions can be specified numerically (octal format) or with a command string. The command string can consist of: the designation for owner (u),

group (g) or other (o); the command to add (+), remove (-) or set (=); the permissions to read (r), write (w) or execute (x); and the commands to set or reset the special flags *set user ID* (s) and *sticky* (t).

Available options

- c** displays the names of files whose permissions actually changed
- f** suppresses error messages
- R** also changes the permissions of files in subdirectories (recursively)
- v** verbosely describes each change

Examples

```
chmod u+x file
    adds execution permission for the owner of the file
chmod go-wx files
    removes read and execution permissions for the
    specified files for the group and other users
chmod g+s file
    sets the set group ID flag of the specified file
chmod =r file
    sets the file's permissions to read-only for everyone
chmod 644 file
    allows read and write permissions for the owner and
    read permissions for all others
```

chown [options] owner[:|.group] files

Changes the owner and optionally the group as well for the specified files. The owner and the group can be specified as numeric IDs or as names.

chown

Available options

- c** displays the names of the files whose owner has actually changed
- f** suppresses error messages

Reference

-
- R** also recursively changes owners of files in subdirectories
 - v** verbosely describes each change
-

cksum

cksum [files]

Computes CRC checksums for the specified files and displays these along with their file size and filename.

clear

clear

Clears the screen.

cmp

cmp [options] file1 [file2]

Compares the contents of two files byte-wise. If the files are identical, 0 is returned; otherwise, 1. If "-" is specified as the filename, then the command reads from the standard input device. The same applies if file2 is not specified.

Available options

- c** displays the characters that are different
 - l** displays the offset and octal values of deviating bytes
 - s** suppresses all screen output
-

comm

comm [options] file1 file2

Compares two line-wise presorted files. Without additional options the output is in three columns: The first column contains the lines that occur only in file1, the second column displays all lines that occur exclusively in file2, and the third column contains all common lines.

Available options

- 1** suppresses column 1
 - 2** suppresses column 2
 - 3** suppresses column 3
-

compress [options] [files]

Compresses the specified files using the Lempel-Ziv method. The compression is indicated by appending ".z" to the filename. All other file attributes are retained.

Available options

- bn** restricts to *n* the number of bits that may be used for coding
 - c** outputs the results to the standard output device and does not change any files
 - f** compresses without confirmation if the target file already exists
 - r** also compresses files in subdirectories (recursively)
 - v** provides a verbose status report
 - V** displays the version number of the program
-

cp [options] file1 file2

cp [options] files directory

Copies file1 to file2 or the specified files into the directory. If the target file (file2) already exists, it is overwritten (although option **-i** requires confirmation).

cp

Available options

- a** combination of **-d**, **-p** and **-r**
- b** creates a backup of files before overwriting them
- d** maintains symbolic and hard links during copying
- f** forces a copy and overwrites existing files
- i** asks for confirmation before overwriting an existing file
- l** creates a hard link rather than a copy of a file
- P** copies files into a target directory hierarchy (which is created if necessary)
- p** also copies the permissions and modification times of the files
- r** recursively copies subdirectories and their contents
- R** see **-r**
- s** creates a symbolic link rather than a copy of a file
- S suffix** changes the extension for backup files to *suffix*

-
- u** prevents the overwriting of a file that has the same name and a newer date
 - v** displays the name of each file on copying
 - x** ignores directories on any file system different from the source file
 - V{numbered, existing, simple}** determines the kind of version control:
 - numbered** always creates a numbered backup
 - existing** creates a numbered backup only for files for which such a backup already exists, and in all other cases creates a simple backup
 - simple** always creates a simple backup
-

cpio

cpio options [arguments]

Copies files into an archive, displays the contents of an archive, or extracts files from an archive. The archives can be on magnetic tape, hard disk or floppy disks. `cpio` has three modes of operation, selected by the options `-i` (*copy in* = unpack) , `-o` (*copy out* = pack) and `-p` (*copy pass* = copy from directories). `cpio` was designed to work with the `find` command

Available options

- 0** accepts filenames terminated with null instead of newline (copy out and copy pass modes)
- a** resets the access times of files that are read so that the reading cannot be discerned from the file date
- A** adds files to an existing archive (with options `-O` or `-F`)
- b** during extraction (copy in), exchanges words and half-words
- B** increases the input/output buffer from 512 to 5120 bytes
- c** uses the (old) portable ASCII format for the file headers
- C *n*** sets the input/output buffer to *n* bytes
- d** automatically creates the necessary subdirectories during extraction
- E *file*** extracts the files whose names are in *file* (copy in)
- f** copies only the files that do not match the specified search pattern

- F *file*** uses the specified *file* as an archive instead of the standard output device. *file* can also contain the name of a host in order to write the archive to a remote magnetic tape (-F zeus:/dev/tape)
- H *format*** reads/writes header information in the specified *format*:
 - bin** old binary format
 - odc** old portable format (POSIX.1)
 - newc** new portable format (SVR4)
 - crc** new SVR4 format with CRC checksums
 - tar** old tar-compatible format
 - ustar** POSIX.1-compatible tar format
 - hpbm** old HP UNIX binary format
 - hpodc** portable HP UNIX format
- i** puts `cpio` in copy-in mode (extraction of an archive)
- I *file*** uses the specified file instead of the standard input device. A host name (zeus:/dev/tape) can be specified, for example, to access an archive on a remote magnetic tape drive
- L** dereferences symbolic links, meaning that not the link but the file to which the link refers is copied
- m** the original modification date of a file is retained on creation of a new file
- M *msg*** enables multivolume archives. If a storage medium is full, the message `msg` is displayed on the screen. The variable `%d` can be used within the message to display the current number of the medium
- n** on display of the directory listing, the UID and GID are displayed as numeric values
- o** puts `cpio` in copy-out mode (creation of an archive)
- O *file*** uses the specified file instead of the standard output device. A machine name (zeus:/dev/tape) can be specified, for example, to access an archive on a remote magnetic tape drive
- p** puts `cpio` in copy-pass mode (copy directories locally)
- r [*user*][:*group*]** changes the file owner in copy-out and copy-pass modes and can only be used by the administrator
- s** exchange bytes in copy-in mode
- S** exchange halfwords in copy-in mode

- t list** displays a *list* of the contents of an archive
- u** permits overwriting of files with the same name and an older version
- v** displays a list of filenames. A verbose version can be obtained by combining with the option **-t**
- V** displays a period (".") for each processed file

Examples

```
find . -name "*.txt" -print | cpio -ocv > /dev/tape
    backs up all files that end with "txt" into an archive on
    magnetic tape
cpio -icdv < /dev/tape
    extracts all files from magnetic tape to the hard disk
find . -print | cpio -pdv /tmp
    copies all files from the current directory to /tmp
```

crontab [-u user] file
crontab [-u user] operations

crontab

Replaces, edits, lists or deletes a user's crontab file. The administrator can process any user's crontab file by using the option **-u**.

Available operations

- e** edits the crontab file in the default editor (environment variable `EDITOR`)
- l** lists a user's crontab file
- r** deletes the crontab file

csh [options] [arguments]

csh

Command interpreter with a syntax based on C.

csplit [options] file [expression]

csplit

Splits the specified file into multiple smaller files and displays the sizes of the generated files. If the specified filename is "-", then data is read from the standard input device. The locations

for the splitting can be specified by an optional expression of the following form:

number	specifies the number of lines after which a new output file is to be created
/regexp/[offset]	regular expression that specifies the splitting locations; an optional positive (+) or negative (-) line offset can be defined
%regexp%[offset]	like the above expression, but in this case the specified section is skipped rather than written to a file
{repetitions}	induces the repeated application of an expression to which it is appended. If an asterisk (*) is specified instead of a number, then the expression is applied until the end of the input file is reached

Available options

- f prefix** specifies the prefix for the generated output files
- b suffix** changes the suffix of the generated files. The format of suffix is based on the format commands of `printf`. `%d` sets the number of the output file in decimal form, while `%x` results in hexadecimal representation
- k** already generated files are preserved, even if the command is aborted
- n n** length of the sequential number in the name of the output files (default 2)
- q** suppresses screen output
- s** see `-q`
- z** suppresses the generation of files of length 0

Examples

```
csplit -k linux.txt '%cut%' {30}
    splits the file linux.txt at positions "cut" into at most
    30 output files
```

```
csplit -k list.txt 10 {100}
    splits the file list.txt into at most 100 files with 10
    lines each
```

ctags [options] files

Reads the specified C, Fortran, Pascal, LaTeX or Lisp source files and generates a list of functions and macros defined therein. This list can be processed in the `vi` or `emacs` editor. A keyword list (tag file) is generated with the name `tags` in the current directory.

Available options

- a** appends the names found to an existing list
- B** generates a search pattern for a backwards search in `vi`
- C** activates C++ mode, where `.c` and `.h` files are treated as C++ code
- d** generates entries for preprocessor definitions as well
- f file** writes the names it finds to *file*. If `-f` is not specified, then the file `tags` is used
- F** generates a search pattern for forward search in `vi` (default)
- H** displays a help text
- i file** continues the search for a tag in the specified *file*
- o file** changes the name of the output *file*
- S** ignores indents
- t** also generates a tag for type definitions
- T** also generates a tag for type definitions, structures, enumerations and C++ member functions
- u** the tag list is updated
- v** generates an index file in `vgrind` format and outputs it to the standard output device
- V** displays the version number
- w** suppresses warnings about duplicate entries
- x** generates a cross reference list in `cxref` format and outputs it to the standard output device

cut options [files]

Cuts a series of fields or columns from a line of the input file. One of the options `-b`, `-c` or `-f` must be specified. Each of these options expects a list that can contain numbers separated by commas or fields defined by hyphens.

Available options

- b *list*** selects the character at the position defined in *list*
- c *list*** selects the columns specified in *list*
- dc** is output together with **-f** to specify the field delimiter character (*c*)
- f *list*** selects the fields (separated by tabulators or the delimiting character) from *list*
- s** restricts output to lines that contain the field separator

Example

```
cut -d: -f1,3 /etc/passwd
    outputs the login names and user IDs of all users
```

date [options] [+format]

date [options] [string]

In the former form, the current date and time are returned in a *format* that can be provided optionally. With the second form, the system administrator can set the system time.

date

Output format:

- %%** percent sign
- %n** new line
- %t** tabulator
- %H** hour (00..23)
- %I** hour (01-12)
- %k** hour (0..23)
- %l** hour (1..12)
- %M** minute (00..59)
- %p** AM or PM
- %r** time in 12-hour format (hh:mm:ss[AM|PM])
- %s** seconds since January 1, 1970, 0:00
- %S** seconds (00..59)
- %T** time in 24-hour format
- %X** time in local format
- %Z** time zone, if defined, else empty
- %a** local abbreviation of day name
- %A** local name of day of week

%b	local abbreviation of month name (Jan...Dec)
%B	local month name (January...December)
%c	local date with time and time zone
%d	day of month (01..31)
%D	date (mm/dd/yy)
%h	identical to %b
%j	sequential day of the year (001..366)
%m	month as number (01..12)
%U	week as number (00..53) where Sunday is the first day
%w	day of week as number (0..6)
%W	week as number (00..53) where Monday is the first day
%x	local representation of the date (dd/mm/yy)
%y	last two digits of the year (00..99)
%Y	year (1995...)

Format of the string to set the time:

DD	day of month
hh	hour
mm	minute
CC	first two digits of year (=century)
YY	last two digits of year
ss	seconds

Available options

-d date	outputs the specified date (which can contain the month name, time zone,... etc.)
-s date	sets the date in arbitrary format (which can contain the month name, time zone,... etc.)
-u	ignores time zone and uses UTC (Universal Coordinated Time)

dd [options=value ...]

copies from the standard input device or a specified file to the standard output device or another specified file. The most frequent options are **if** to specify the input file and **of** to specify the output file. **dd**, for example, can be used to write a kernel image file directly onto a diskette or to make a boot diskette from a disk image.

dd

Available options

bs=n	sets the block size for input and output to <i>n</i> bytes. Optionally, <i>n</i> can be specified with units, e.g., 8k for 8 kilobytes
cbs=n	determines the size of a field in converting to bytes
conv=flags	converts the input according to the following arguments:
ascii	EBCDIC to ASCII conversion
ebcdic	ASCII to EBCDIC conversion
ibm	ASCII to IBM EBCDIC conversion
block	converts variable-length fields to fields of length <i>cbs</i> and fills the spaces with blanks
unblock	converts fixed-length fields (<i>cbs</i>) to variable-length fields
lcase	converts upper-case to lower-case letters
ucase	converts lower-case to upper-case letters
swap	swaps every two bytes of the input file
noerror	ignores errors during reading
notrunc	does not truncate the output file
sync	fills spaces in the input blocks of size <i>ibs</i> with zeros
count=n	copies only <i>n</i> blocks
if=file	specifies the input file
of=file	specifies the output file
ibs=n	sets the size of the input buffer
obs=n	sets the size of the output buffer
skip=n	skips <i>n</i> blocks of input

df [options] [paths]

Outputs the number of occupied and free blocks of file system. If no path is specified, then a list of all current file systems is output. If a path is specified, an overview is provided of the associated file systems. Alternatively, the direct path of a device (/dev/hda1) on which the file system is located can be specified. Normally only real file systems with a storage volume greater than zero are output.

df

Available options

- a** displays all current file systems, including those of size zero
- i** instead of block information, displays i-node statistics
- k** uses a block size of one kilobyte (default)
- P** uses POSIX output format
- t *type*** restricts output to file systems of a certain type
- x *Type*** ignores file systems of a certain type during output

diff

diff [options] file1 file2

Compares two files or all files in two directories. If one of the two paths is specified as "-", then the files are expected from the standard input device. The output of `diff` lists all lines that occur in only one file or that are different. This output can be used by `patch` to make changes in files. Another alternative for comparing or merging files is the Emacs Lisp program `ediff`.

Available options

- a** treats all input files as text files and compares linewise
- b** ignores differences in the number of blanks (at the end of a line as well)
- B** ignores blanks
- c** generates output with three lines of context around each difference
- C*n*** like `-c`, but *n* lines of context are output around each difference
- d** uses a better, although slower, algorithm for file comparison
- D *name*** mixes the two files and inserts appropriate preprocessor instructions (`#ifdef name`) to make the two versions distinguishable. If *name* is defined during compilation, then the version is output to `file1`, otherwise to `file2`
- e** outputs instructions for the `ed` editor to be able to generate `file2` from `file1`
- f** like option `-e`, but reversed, yet it cannot be used as an `ed` script

-
- h** is ignored
 - H** uses heuristics to increase the speed
 - i** ignores differences in upper/lower case
 - l** (only when comparing whole directories) the output can be processed with the command `pr` so that each file begins on a new page
 - n** generates output in RCS format
 - N** in comparing two directories, missing files are considered as existing, but empty
 - q** simply reports whether the files are different
 - r** (only when comparing whole directories) subdirectories are handled recursively and all files are compared
 - s** reports whether two files are identical
 - S *file*** starts comparing directories with a certain file
 - t** replaces tabs with blanks
 - T** outputs a tab instead of a blank at the start of every output line
 - u** generates output in GNU-specific "unified" format
 - v** displays the version number
 - w** ignores blanks and tabs when comparing lines
 - x *pattern*** ignores files and subdirectories that match the specified *pattern* (when comparing whole directories)
 - y** outputs in easy-to-read, two-column format
-

diff3 [options] file1 file2 file3

Compares three files linewise.

diff3

Available options

- a** linewise comparison treating all input files as text files
- A** inserts all changes between file2 and file3 in file1 and marks conflicts
- e** generates a script for the `ed` editor that integrate all changes from file2 to file3 in file1
- E** like option `-e`, but the output is less verbose
- i** generates `w` and `q` commands at the end of a generated `ed` script
- m** applies the edit script to file1 and displays it

Reference

- T outputs a tab instead of a blank at the start of every output line
- v outputs the version number of the command
- x like option -e, but only overlapping changes are output
- X like option -E, but only overlapping changes are output
- 3 like option -e, but only non-overlapping changes are output

dirname

dirname pathname

Extracts the directory part of a complete path specification (counterpart to `basename`). If the path does not contain a file at the end, then "." is returned.

du

du [options] [files | directories]

Outputs the sizes of the specified files or directories.

Available options

- a outputs the sizes of all files, not just directories
- b outputs the file size in bytes
- k outputs the file size in kilobytes
- l outputs the sizes of (hard) linked files, even if this means handling them doubly
- s outputs the total size of all files and subdirectories
- x ignores directories in different file systems

echo

echo [-n] [text]

This command is usually built into shells. It outputs `text` to the standard output device. The option `-n` suppresses the output of the newline character.

ed

ed [options] [file]

This is an antiquated standard editor which, apart from its use with the `diff` command, is no longer of any importance.

egrep

See grep.

env [options] [variable=value] [command]

If invoked without parameters, this command produces a list of all environment variables. In addition, this command permits starting commands in a modified environment. In the command line, new variables can be defined or existing ones removed.

Available options

- i** ignores the inherited environment
 - u name** removes the specified environment variable
-

expr arg1 operator arg2 [operator arg3 ...]

Evaluates an expression and outputs the result to the standard output device. Expressions can be numeric, logical or relational. This command is usually used in shell scripts.

Arithmetic operators:

+, -, *, /, % (modular rest)

Relational operators:

=, !=, >, >=, <, <=

Logical operators:

| (or), **&** (and),
: (seek Arg2 as a regular expression in Arg1)

Examples

expr 7 + 8 / 2

evaluates to 7 (integer arithmetic, left to right!)

expr \$s = "hello"

evaluates to 1 if *s* contains the string "hello", else 0

Reference

false

false

This command does nothing and returns *false* (not 0). (Also see *true*.)

fdformat

fdformat [-n] device

Executes low-level formatting of a diskette. The required parameter is the path of the corresponding device. The first drive is addressed as `/dev/fd0XXX`; the second, as `/dev/fd1XXX`. The `-n` option suppresses subsequent verification of the diskette.

Device	Sectors	Tracks	Size	Capacity (kB)
<code>/dev/fd0h1200</code>	15	80	5 ¼	1200
<code>/dev/fd0D720</code>	9	80	3 ½	720
<code>/dev/fd0H1440</code>	18	80	3 ½	1440
<code>/dev/fd0E2880</code>	36	80	3 ½	2880

fgrep

fgrep

See grep

file

file [options] files

Outputs the types of specified files. The file type is recognized on the basis of an extensible rule file (`/etc/magic`).

Available options

- c** for checking the rule file
- f file** examines the files listed in *file*
- m file** uses the specified rule *file* instead of `/etc/magic`
- L** also follows symbolic links
- z** enables the processing of compressed files

find

find pathname constraints

This command recursively searches in directories for files that meet all the specified constraints. The list of constraints

is evaluated from left to right. Individual constraints can be negated by a preceding exclamation mark (!). An OR conjunction between two expression is defined with `-o`. `find` is particularly useful in combination with other commands (e.g., `cpio`).

Numeric specifications can be represented in three ways:

- `+n` value larger than `n`
- `n` value equal to `n`
- `-n` value smaller than `n`

Available options (always true)

- depth** files contained in a directory are processed before the directory itself
- follow** also branches to directories indicated by symbolic links (follows symbolic links)

Possible constraints

- amin *n*** files accessed in the last *n* minutes
- anewer *file*** files accessed more recently than the specified *file*
- atime *n*** files that were last accessed *n* days ago
- ctime *n*** files that were last modified (either the file itself, the permissions or the owner) *n* days ago
- fstype *type*** files in a certain *type* of file system (e.g., `ext2`, `msdos`, `proc`)
- group *group*** files belonging to a certain *group* (name or ID)
- inum *n*** files with i-node number *n*
- links *n*** files that possess *n* links
- local** files physically stored on the local system
- mtime *n*** files that were last modified (only the file itself) *n* days ago
- name *pattern*** the names of the files match the specified wildcard *pattern*
- newer *file*** the last modification of the files must be more recent than the specified *file* (see also `mtime`)
- nogroup** files whose group does not exist in `/etc/groups`

-
- nouser** files whose owner does not exist in /etc/passwd
 - perm *nnn*** the permissions of the files must match the octal representation *nnn*
 - size *n[c,k]*** files of size *n* blocks, *n* bytes or *n* kilobytes
 - type *c*** files of type *c*, where *c* must derive from the following list:
 - b** block special file
 - c** character special file
 - d** directory
 - p** FIFO or named pipe
 - l** symbolic link
 - f** normal file
 - user *user*** files belonging to a certain *user* (name or ID)

Possible actions

- exec *command* {} \;** executes the *command* for each file and tests whether the return code is 0. During execution {} is replaced by the name of the current file
- ok *command* {} \;** like exec, but the user must confirm the *commands* with "y"
- print** outputs files or directories found
- printf *format*** like -print, but the format of the output can be influenced by a *format* string

Examples:

```
find . -type f -print
    outputs all normal files in the current directory and its
    subdirectories
find . /usr/include -type f \
    -exec grep "read" {} \; -print
    searches all normal files in /usr/include for the
    character string "read"
```

finger [options] [user]

Provides information on users. *user* can be in the form of name, name@host or @host. In the first two forms the names of the

users, the times, the last logins, and additional information are output. If the file `.plan` or `.project` exists in the user's home directory, then it is also displayed. If only one host is specified with `@host`, then all users are listed who are currently logged into that system.

Available options

- l** forces verbose output (at `@host`)
- m** the specified `user` must exactly match the user name. Without this option the specified name is also compared with the full name of the user as stored in the file `/etc/passwd`
- p** the files `.plan` and `.project` of the respective user are not displayed
- s** forces brief output format

ftp [options] [host]

This program is for transferring files with the ftp protocol. The `host` can be specified by name or IP address. If no `host` is specified, then the program responds with a prompt that allows entry of ftp commands. Entering `help` evokes a help text.

Available options

- d** debug mode
- g** toggles off the use of wildcards for filenames
- i** turns off queries (`mget`, `mput`)
- n** suppresses automatic login on computer listed in the file `.netrc`
- v** displays all the ftp server's messages

gcc[options] [files]

In addition to C, C++ and Objective C, the GNU C compiler supports back ends for other languages such as Ada and Pascal. A more detailed description of these features can be found in the GNU Info documents.

ftp

gcc

grep [options] regexp [files]

Searches *files* or data from the standard input device linewise for a regular expression (*regexp*) and outputs the found lines to the standard output device.

Available options

- b** additionally outputs the byte position where the expression was found
- c** outputs only the number of lines in which the expression was found
- h** suppresses the output of filenames
- i** ignores differences in size and (upper/lower) case
- l** outputs only the names, but not the lines, of files in which the expression was found
- n** outputs the line numbers of found lines
- s** suppresses error messages in case a file does not exist or cannot be opened
- v** searches for lines that do not contain the regular expression

groff [options] [files]

groff is the GNU variant of *nroff* and *troff*. The command serves to format Manual pages and other documents that are available in the appropriate format. Additional preprocessors such as *eqn* or *tbl* are integrated in *groff* and can be activated with options. The results can be stored in ASCII, DVI or PostScript format. In formatting documents, it is important to specify the macro package used. For Manual pages, for example, the option *-man* would be specified. The formatted file is written to the standard output device.

Available options

- a** outputs pure ASCII format
- e** activates the *eqn* preprocessor
- E** suppresses error messages
- h** outputs help text
- m *macro*** uses a special *macro* package for formatting

-man macros for Manual pages
 -ms ms macro package
-p activates `pic` preprocessor
-s activates `soelim` preprocessor
-t activates `tab` preprocessor
-T *format* specifies output *format* (`ascii`, `ps`, `dvi`) and
-v outputs the version number

Example

```
zeus:/home/uhl> groff -man -Tps ls.1 > ls.ps
```

formats the Manual page `ls.1` with the corresponding macro package and outputs the result in PostScript format to the file `ls.ps`

groups [user]

Outputs the groups to which the specified `user` belongs. A parameterless invocation lists all the current user's own groups, and with a parameter the groups of the specified `user`. The command evaluates the files `/etc/passwd` and `/etc/groups`.

gzip [options] [files]

Compresses or decompresses `files` using the LZ77 method and adds the extension `.gz` to the filename. If no file (or `-`) is specified, then the standard input device is read and output goes to the standard output device. `gzip` can also decompress files packed with `compress` (ending in `.Z`).

Available options

-a adapts the end of line in ASCII texts to the respective system (CRLF or LF)
-c writes the results to the standard output device without overwriting the input file
-d decompresses packed files
-f forces an overwrite of existing files
-l for a compressed file, displays its size in packed and unpacked form, the compression rate, and the name of the original file

groups

gzip

Reference

-
- q** suppresses warnings
 - r** recursively works through subdirectories
 - S *.suffix*** changes the file *suffix* for compressed files
 - v** displays the name and the compression rate for each file
 - #** sets the quality of compression from 1 (poor) to 9 (good), where default compression is 6
-

head

head [options] [files]

Displays the first ten lines of the specified (text) files. On specification of multiple files, the filename precedes the file contents in the output.

Available options

- #** changes the number of lines to be output to the specified value
 - c *n[b|k|m]*** outputs the first *n* bytes, where the specification can be made in bytes (*b*), kilobytes (*k*) or megabytes (*m*)
 - q** suppresses the output of filename
-

hostname

hostname [name]

Parameterless invocation displays the name of the host; otherwise the specified name is set. The host name is usually set on system startup and requires superuser permissions.

id

id [options]

Displays the real and effective user ID (UID) and all groups (GID) of the current user.

Available options

- g** displays only GID
- G** displays only the additional groups to which a user belongs
- n** displays GID or UID as name (only in combination with options **-g**, **-u**, **-G**)

-
- r** displays the real instead of the effective GID (only in combination with options `-g`, `-u`, `-G`)
 - u** displays only UID
-

join [options] file1 file2

Joins two alphabetically sorted ASCII files via a key. Lines with identical keys are joined and written to the standard output device. Keys must be separated by blanks or tabs. If no further options are specified, the first column is used as the key.

join

Available options

- a [n]** adds an empty line to the output if one line of file *n* (1 or 2) does not have a matching key in the other file
 - e *string*** replaces empty output fields with the specified character *string*
 - j *n m*** uses column *m* of file *n* (1 or 2) as the key
 - o *n.m*** displays only column *m* of file *n*
 - t *z*** uses character *z* as field delimiter (input/output)
-

kill [options] processes

This command is usually built into a shell. It sends a signal to one or more processes. Without further options, a `TERM` signal is sent, which orders a process to terminate. Only the system administrator can send signals to processes that she/he does not own. The processes are specified with their process number (PID). The signals can be specified numerically or symbolically.

kill

Available options

- l** lists all signal names
- signal** sends a certain *signal* to the specified processes. The following signals are useful in this context:

No..	Name	Explanation
1	SIGHUP	Generated on interruption of a terminal connection. For many daemons it serves to read the configuration files anew

2	SIGINT	Equivalent to entering <Ctrl-c>
3	SIGQUIT	Terminates a process and invokes a core dump
9	SIGKILL	Terminates a process. This signal cannot be intercepted
15	SIGTERM	Terminates a process (default).
10	SIGUSR1	User-specific signal whose meaning is different in each application
12	SIGUSR2	See SIGUSR1

ksh

ksh [options] [arguments]

See bash.

last

last [options] [attribute]

Provides information from the login statistics (/etc/wtmp). Without additional arguments, it outputs a list of all login, logout, shutdown and reboot activities. This list contains the name of the user or the event, the login terminal, the login host and the time. A selection can be limited to certain entries by specifying search attributes (name, login terminal).

Available options

- #** limits output to a certain number of lines
- f file** uses the specified *file* instead of /etc/wtmp as the data base
- t terminal** lists only logins entered from a particular *terminal*
- h computer** lists only logins entered from a certain *computer*

Example:

```
hermes:/root# last uhl
uhl  tty4    mobby    Sun Jan 29 17:24  still logged in
uhl  tty2     tonne    Sun Jan 29 16:32 - 16:47 (00:15)

wtmp begins Sun Jan 29 15:18
hermes:/root#
```

Reference

ld [options] object_files

The linker links individual object files to an executable program. It is seldom invoked directly. Normally the C compiler or the `make` command automatically invokes the linker.

ld

ldd [options] [programs]

Lists the dynamic libraries that a program needs.

ldd

Available options

- d** carries out a relocation and lists missing functions (only ELF format)
 - r** carries out a relocation for data and program code and lists missing objects (only ELF format)
 - v** outputs the version number of the command
 - V** outputs the version number of the dynamic linker (ld.so)
-

lex [options] [files]

The scanner generator creates the output file `lex.yy.c` from a scanner grammar as input file.

lex

ln [options] path target_path

Creates a link. Without options, it creates a hard link to a file. With the option `-s` a symbolic link is created that could also point to a directory. If `target_path` already exists and is a file, then an error message is output. Only with the option `-f` is this file overwritten. If `target_path` is a directory, then the links are created in this directory.

ln

Available options

- f** any existing files are overwritten without confirmation
 - s** symbolic links are created
-

lpc [command [argument]]

Serves to control printer spoolers. It enables the activation and deactivation of individual printers and their printing queues,

lpc

shifting printer jobs within the printing queues, and outputting status information. Invoking `lpc` without an argument produces an interactive command modus. Alternatively, these commands can also be passed to `lpc` on invocation.

Available commands

help	displays a list of available commands
abort {all printer}	terminates active spooler(s) and disables the corresponding <i>printer</i> (s)
clean {all printer}	removes all incomplete files from the specified <i>printer</i> queue(s)
disable {all printer}	disables the corresponding <i>printer</i> (s)
down {all printer} message	turns off the specified queue, disables <i>printer</i> (s) and writes the specified <i>message</i> in the printer status file. This message is output on invocation of <code>lpq</code>
enable {all printer}	enables the specified printer queue(s) and permits the addition of new jobs
exit, quit	ends the <code>lpc</code> program
restart {all printer}	attempts to restart <i>printer</i> daemon(s)
start {all printer}	activates <i>printer</i> (s) and starts <i>printer</i> daemon(s) for the specified <i>printer</i> (s)
status {all printer}	outputs status information on currently active <i>printer</i> daemon(s) and queue(s)
stop {all printer}	stops the <i>printer</i> daemon on completion of the current job and disables the <i>printer</i>
topq printer [job#] [user]	places the specified job at the head of the queue
up {all printer}	activates queue(s) and starts <i>printer</i> daemon(s)

lpq [options] [job#s] [user]

Provides information on the current status of printer queues.

Available options

-l verbose status report on each job

-P name selects a printer queue

lpr [options] [files]

Sends *files* to a printer queue. Alternatively, data can be printed via the standard input device. Invocation without options outputs to the queue *lp*.

Available options

-#n creates *n* copies of the specified documents
-C text prints a job classification on the title page
-h suppresses the output of a header before a print job
-J job prints a *job* name on the title page
-m sends a mail to the user on completion of the job
-P name selects the specified printer queue
-r deletes the file after printing (with option *-s*)
-s file is not spooled but linked. Thus the printer file must not be deleted during printing
-U user prints the *user* name on the title page

lprm [options] [job#s] [user]

Removes entries from a printer queue. *Job numbers* or *user* names can be specified as selection criteria. If no argument is specified, the active job is removed.

Available options

- removes all entries from a queue
-P name selects the specified printer queue

ls [options] [files]

Displays the contents of directories or lists specific *files*. If no *files* are specified, the contents of the current directory are listed. If *files* are specified, then only the such files are listed that match the filename (with wildcards).

Available options

lpr

lprm

ls

-
- a** displays all files, including those beginning with a period
 - A** like option **-a**, but suppresses the entries "." and ".."
 - B** ignores backup files that end in tilde (~)
 - b** displays nonprintable characters as octal numbers
 - c** sorts files by time of last status change
 - C** displays only filenames, but in multiple columns (default)
 - d** on specification of a directory name, lists only the directory itself, not its contents
 - f** unsorted output
 - F** appends a special character to each filename to indicate the file's type (normal file, directory, executable file, link, ...)
 - G** suppresses the output of group in long format
 - i** displays the associated node for each file
 - k** displays file size in kilobytes
 - l** long format displays every file in a line along with its permissions, owner, group, size, etc.
 - L** for symbolic links, shows the file or directory to which the link points rather than the link itself
 - m** lists filenames linewise, separated by commas
 - n** lists UID and GID numerically
 - r** lists files sorted backwards
 - R** recursively lists subdirectories and their contents
 - s** lists file size in kilobytes before the filename
 - S** sorts list by file size
 - t** sorts list by date of last modification, with newer files coming first
 - u** sorts list by time of last access
 - x** lists files in horizontally sorted columns
 - X** lists files sorted by file extension
-

m4 [options] [files]

m4

This macro processor is used for various program files, in the GNU Autoconf system, and for `fvwm` configuration files.

mail [options] [addresses]

This program is for reading and sending e-mail. Users should rather use the program `pine` or a graphical mail reader. However, `mail` proves superb for simply sending text files because the contents can be transferred via the standard input device.

Example:

```
mail linux@fh-heilbronn.de <critique.txt
```

make [options] [targets]

Reads a makefile and updates one or more targets. `make` is usually used for compiling of source files.

Available options

- C *directory*** changes to the specified *subdirectory* before a makefile is read
 - d** provides additional debugging information
 - e** environment variables overwrite corresponding variables in the makefile
 - f *makefile*** uses the specified *makefile*
 - I *directory*** searches in the specified *directory* for imported makefiles
 - k** on error, aborts only the current target, not the complete make process
 - n** only outputs commands without executing them
 - p** outputs internal macro definitions
 - r** uses no default rules
 - s** suppresses screen output
 - t** provides *files* to be processed with the current date without executing the corresponding operation
 - w** displays the current working directory before and after executing an operation
-

Reference

man

man [options] [[section] name]

Displays Online Manual pagewise on the screen. These pages are located in a subdirectory under `/usr/man` or in other directories listed in the environment variable `MANPATH`.

Available options

- a** displays all Manual pages that match the specified name
- f** equivalent to the command `whatis`
- h** displays a help page
- k** equivalent to the command `apropos`
- M path** specifies a list of additional directories in which to search for Manual pages (see `MANPATH`)
- w** displays not the contents but the access path of a Manual page

mesg

mesg [y | n]

Determines whether other users can write messages on the terminal with `write`. If `mesg` is invoked without options, the current status is displayed.

mkdir

mkdir [options] directories

Creates directories.

Available options

- m perms** creates a new directory with the specified *permissions*
- p** if a directory path is specified where individual subdirectories do not exist, then these are created also

more

more [options] [files]

Displays files by (screen) page. `<Enter>` scrolls one line down and the space bar advances to the next screen page. `<h>` displays help with all commands and `<q>` quits the `more` command. If no file is specified, then `more` reads from the standard input device.

Available options

+#	begins with the specified line number
-d	displays the message "Press space to continue, 'q' to quit" at the end of a screen page
-f	counts logical rather than screen lines for page breaks and counts broken lines only once
-l	ignores form feed control character (^L)
-s	suppresses the output of multiple neighboring blanks
-u	suppresses underlining

mttools

This is a group of commands that permit simple access to MS-DOS file systems. Normally these are used to handle diskettes. Note that access to a DOS partition of a hard disk is simpler if it is mounted (see `mount`). The individual commands largely correspond to the DOS commands. This means that floppy disk drives can be accessed with DOS's usual letter designations (A:, B:) if the drives were correctly configured in the file `/etc/mttools`.

Commands

mattrib	modifies file attributes
mcd	changes the current directory
mcopy	copies files
mdel	deletes files
mdir	displays a directory listing
mformat	formats a low-level formatted diskette with a DOS file system
mlabel	changes the volume label
mmd	creates a subdirectory
mrdd	removes a subdirectory
mren	renames a file
mtype	displays the contents of a file

mount [options] [device] [mount_location]

Links new file systems into the directory tree. A file system is attached to the UNIX file tree at a defined `mount location`.

mttools

mount

Unspecified parameters are taken from the entries of the file `/etc/fstab`.

Available options

- a** automatically mounts all file systems specified in `/etc/fstab`
- f** suppresses the actual `mount` system invocation (practical with option `-v`)
- n** suppresses entries in `/etc/mtab`
- o *opts*** additional *options* that depend on the respective file system

General options

- async** all input and output is asynchronous
- auto** the file system can be mounted with the `-a` option
- defaults** standard options: `rw, suid, dev, exec, auto, nouser, async`
- dev** permits the use of character- and block-oriented devices
- exec** permits execution of commands
- noauto** can only be mounted explicitly, but not with the option `-a`,
- nodev** suppresses the use of character- and block-oriented devices
- noexec** suppresses the execution of commands
- nosuid** SUID and SGID bits have no effect
- nouser** forbids a normal user to mount file systems
- remount** permits remounting of a file systems, e.g., to change mount options
- ro** mounts the file system as read-only. This option must be specified to mount CD-ROM file systems
- rw** mounts the file system for reading and writing
- suid** enables the execution of SUID and SGID commands
- sync** all input and output operations are synchronous
- user** permits a normal user to mount the file system

File-system-specific options

- case={*lower* | *asis*}** (hpfs) sets (upper/lower) case sensitivity

check=value (ext2) enables the choice of consistency checks before mounting a file systems

- none** no consistency checks
- normal** check i-node and block bitmap (default)
- strict** also checks consistency of free blocks

check=value (msdos) determines the form for specifying file names

- relaxed** case insensitive, long file names truncated
- normal** special characters (*, ?, <, ...) not accepted (default)
- strict** no long file names and no special characters

conv=value determines whether end-of-line (EOL) character is converted on access to file system (msdos, hpfs, iso9660)

- binary** no EOL conversion (default)
- text** CRLF/LF conversion for all files
- auto** no conversion on files with the following extensions: exe, com, bin, app, sys, drv, ovl, ovr, obj, lib, dll, pif, arc, zip, lha, zoo, tar, z, arj, tz, taz, tzp, tpz, gif, bmp, tif, gl, jpg, pcx, tfm, vf, gf, pk, pxl, dvi

block=value specifies block size for iso9660 file systems

cruft sets the *cruft* flag to overcome an error in certain CD-ROM premastering programs (iso9660)

debug creates debug messages (ext2, msdos)

errors=value determines error handling (ext2)

- continue** no special error handling (default)
- remount ro** file system is remounted read-only
- panic** on error, force a kernel panic

fat=value overwrites the automatically detected *value* for the FAT type (available values being 12 and 16) (msdos)

gid=value establishes the GID for each file of the file system (msdos, hpfs)

grpuid new files receive the same GID as the directory in which they are created (ext2)

nocheck equivalent to *check=none* (ext2)

nogrpuid new files receive the GID of the creating process, as in System V (ext2, default)

norock turns off Rockridge extensions, ending case sensitivity and long filenames (iso9660)

quiet suppresses corresponding error messages on attempts to execute the commands `chmod` and `chown` (msdos)

sb=value uses an alternative superblock at the specified block position (normally at positions 1, 8193, 16385, ...) (ext2)

sysvgroups see `nogrpuid`

uid=value determines the GID for each file in the file systems (msdos, hpfs)

umask=value determines the umask for files (msdos, hpfs)

-r file system is mounted as read-only

-t type mounts a file system of a certain *type* (default: minix; possible values: minix, ext, ext2, xiafs, msdos, hpfs, proc, nfs, iso9660, sysv, xenix, coherent)

-v outputs verbose messages

mv [*options*] *path target*

mv

Moves files and directories or renames them. If the target already exists and it is a file, it is overwritten; if it is a directory, the specified files and directories are moved into the existing directory. If the *target* does not exist, then only a file or a directory can be specified as the source, and it is renamed to the *target name*.

Available options

-b creates a backup of a file before overwriting it

-f does not ask for confirmation before overwriting files

-i asks for confirmation before overwriting files

-u moves a file only if it is newer than a target file of the same name

-S see `cp` command

-V see `cp` command

nice [-n value | -value] commands [arguments]

Executes commands with a higher nice level, i.e., a lower priority. `nice` is usually integrated in the shell. The maximum nice level is 19. The system administrator can also specify negative values to -20. A default nice-level of 10 is used in lieu of a specification.

nice

nm [options] files

Outputs the symbol table of object files or libraries.

nm

nohup command [arguments] &

This command is usually integrated into the shell. It prevents termination of the shell when the specified command terminates.

nohup

nroff [options] files

Formats files that contain corresponding format statements for output on the screen or the printer (see also `groff`).

nroff

openwin

Shell script to start the X11 environment.

openwin

passwd [user]

Changes the user's own password. The system administrator can also change the passwords of other users.

passwd

pr [options] [files]

Prepares text files for printing. The file contents are prepared pagewise and provided with a title line containing the date, the filename and the page number.

pr

Available options

+page begins printing starting with the specified *page*
-column produces multiple-*column* output

-
- a** prints columns alongside rather than under one another
 - c** outputs nonprintable characters in "^" notation
 - d** double-spaced printout
 - e[chars[width]]**
replaces any number of *characters* with a number of blanks. The default value for tab characters is 8 blanks
 - f** produces a form feed at the end of a page instead of generating a series of blank lines
 - h text** replaces the filename in the title line with the specified *text*
 - i[chars[width]]**
reverse effect of option **-e**
 - l length** determines the *length* of a page (default 66 lines)
 - n[chars[width]]**
outputs a sequential number before each line; optionally, a *character* that separates the number from the text and the *width* of the number can be specified
 - o width** creates a left margin of specified *width*
 - r** suppresses error messages for files that cannot be opened
 - t** suppresses the header and footer
 - v** outputs nonprintable characters in format
 - w** specifies page width in characters (default 72)
 - x** displays the processes that are not assigned to a terminal
-

ps [options]

ps

Outputs a list of currently active processes.

Available options

- a** displays the processes of all users
- h** suppresses the header line
- j** outputs the process' group ID and session ID
- l** verbose output format
- m** provides an overview of storage allocation
- r** lists only currently running processes
- s** provides information on signal status
- u** outputs the name of the process owner and the start time

-w suppresses the truncation of command lines for wide output

pwd

Outputs the complete path of the current directory.

pwd

rcp [options] sources target

Copies files between computers. The sources and the target are in the form `user@host:path`, whereby `user@` can be omitted, in which case the current user name is used. For local files only the path is specified.

rcp

Available options

-r recursively copies subdirectories and their contents
-p retains the file attributes (date, permissions) during copying

rlogin [options] host

Similar to `telnet`, this command provides a connection to the specified `host` and logs in there. If the current user is entered on the remote host in the files `.rhosts` or `/etc/hosts.equiv`, then no password is required.

rlogin

Available options

-l name uses *name* as the user name on the remote host

rm [options] files

Removes one or more files. Removing a file requires write permission in the containing directory. If the file is write protected, confirmation is required. Directories are removed with `rmdir`.

rm

Available options

-f removes files, even if write protected, without confirmation
-i asks for confirmation for each file

Reference

rmdir

- r** recursively removes subdirectories and their contents
- v** displays each filename on removal

rmdir [options] directories

Removes subdirectories. A directory must be empty to be removed. Alternatively, `rm` can be used with option `-r`, which removes subdirectories and their contents.

rsh

rsh [options] host [commands]

Executes commands on a remote host. Access must be permitted via an entry in `/etc/hosts.equiv` or `~/.rhosts`.

Available options

- l user** attempts to execute the specified command under another user name
- n** redirects the standard input device to `/dev/null` (works around problems with `csh`)

Example

```
rsh -l uhl zeus.demo.de ls
      executes the command ls as user uhl on the host
      zeus.demo.de
```

sdiff

sdiff [options] file1 file2

Compares two files and outputs the differences in two columns (also see `diff`). This output is easier to read than that of `diff`. Lines that are not contained in one of the two files are marked with `"<"` or `">"`. A pair of lines that differ are indicated with `"|"`.

sed

sed [options] [files]

Modifies files without interaction with the user. This command is usually used in shell scripts to replace, delete or insert text. If no file is specified, then `sed` works with the standard input device.

Available options

-e 'statements'	executes the edit statements with the specified files
-f script file	reads the edit statements from <i>script file</i>
-n	suppresses the echo of input lines on the screen

shutdown [options] time [message]

Changes the run level of the system or terminates the system. A time and a warning message can be passed as arguments. For an immediate shutdown, the time now is specified.

shutdown

Available options

-c	interrupts a shutdown in progress
-h	halts the system with the termination of all processes and unmounts the file systems
-k	does not execute a shutdown, but only displays the warning
-r	reboots the system
-t sec	delay in seconds between the display of the warning message and sending the kill signals

sleep time

Sleeps the specified time in seconds. This command is usually used in shell scripts.

sleep

sort [options] [files]

Sorts the lines in the specified files. If no files are specified, the standard input device is processed.

sort

Available options

+n-m	sets the sorting key between fields <i>n</i> and <i>m</i>
-b	suppresses leading blanks
-c	checks whether the specified files are already sorted; if so, the program terminates with an error message
-d	ignores punctuation marks during sorting
-f	case-sensitive (upper/lower)
-i	ignores nonprintable ASCII characters

-
- m** mixes two input files
 - M** interprets the first three characters as a month name (JAN, FEB, ..., DEC) and sorts by month
 - n** sorts numerically
 - o file** redirects the standard output device to a file
 - r** inverts sorting order
 - t char** specifies the delimiting character for columns (default: blank or tab)
 - u** removes duplicate lines

Example

```
sort +2n -t: /etc/passwd
    sorts the password file numerically according to the
    third column
```

strings

strings [options] files

Searches for character strings in binary and object files or programs. A character string is considered to be any sequence of four or more printable characters terminated with null.

Available options

- a** normally for object files, only the code and data segments are searched; this option assures that the whole file is processed
 - f** each character string is preceded by the corresponding filename
 - n** specifies the minimum length of the character string (default: 4)
 - o** outputs the position of a character string in bytes
-

strip

strip [options] files

Strips symbol, debug, line numbers, and other information from object files and programs, thus reducing their size.

stty [options] [modes]

Sets terminal IO modes. This includes all general settings of the terminal as well as speed and handshaking and the function of special characters. A list of all possible settings is displayed using the option `--help`.

Available options

-a displays all current settings
--help displays help text

su [-] [user] [arguments]

Starts a new shell as a different user. This program is used to log in on a terminal that is already being used by another user. Omitting the user opens a root shell. The new shell is terminated by entering `exit` or `<Ctrl-d>`. If `"-"` is specified as option, then the complete login process is run on opening the new shell. In addition, the option `-c` allows execution of commands under a different user ID.

tail [options] [files]

Outputs the last ten lines of the specified files.

Available options

-c [b|k|m] outputs the last *n* bytes in blocks (*b*), kilobytes (*k*), or megabytes (*m*)
-f does not terminate after outputting the last lines, but waits until the file is written to. As soon as new lines are appended to the file, they are output. In this mode the program is terminated with `break` (`<Ctrl-c>`). This mode proves especially suitable for monitoring log files
-n outputs the last *n* lines
-v outputs the filename as title line

Reference

talk

talk user[@host] [tty]

Sets up a `talk` connection to the specified user. If this user is logged in on multiple terminals, then the terminal to be used can be entered in the command line. A `talk` connection splits the terminal screen in two parts, with local input shown in the top half and remote input in the lower half. The connection is terminated with `<Ctrl-c>`. Unfortunately, there are two incompatible versions of `talk`, so that connection to a different platform does not always succeed.

tar

tar [options] [archive] [files]

Manages `tar` archives (originally on magnetic tape). This command writes files to an *archive* or reads them from an archive. At least one of the following operations must be passed as parameter.

Available operations

- c** creates the archive
- r** appends files to an archive (not on tape)
- t** outputs the contents of an archive
- u** appends files to an archive if they are not already contained or if they have been modified (not on tape)
- x** extracts files from an archive

Additional options

- b *n*** sets the blocking factor to *n*
- f *archive*** specifies the *archive*, which can be a normal file or a device file such as `/dev/rmt0` for a tape drive or `/dev/fd0` for a diskette
- h** archives referenced files instead of their symbolic links
- k** prevents overwriting of existing files
- L** follows symbolic links
- m** sets the modification time on extracting files to the current time
- M** creates or extracts from a multivolume archive, which can encompass multiple diskettes or tapes

-
- N *date*** archives only files that are newer than the specified *date*
 - o** sets the owner on extracting files to the current user
 - O** extracts files to the standard output device
 - v** displays the filename on archiving or extracting
 - z** compresses the archive on creation and decompresses on extraction

Examples

```
tar -cvf archive.tar *
    saves all files and subdirectories of the current directory
    in an archive named archive.tar
tar -cvf /dev/fd0 *.txt
    saves all files in the current directory with the extension
    .txt from the diskette in the first floppy disk drive
tar -xvfb 20 /dev/rmt0
    extracts all files from the first tape drive (block size 20)
tar -tvfz archive.tar.z
    lists the contents of a compressed tar archive
```

tee [options] [files]

This program is used as a filter. It copies the standard input device to the standard output device and the specified files.

Available options

- a** appends the data received from the standard input device to the end of the file instead of overwriting itself
- i** ignores interrupt signals

telnet [host [port]]

Opens a connection to the specified host using the telnet protocol. A port number can be specified optionally. This program is often used to test services that are available for connections on certain ports. If no host is specified, then telnet goes into command mode, where telnet commands can be entered. The command `help` lists all important commands.

tee

telnet

test condition

Evaluates the specified `condition` and returns zero if the result is true, else a nonzero value. Alternatively, the `condition` can be put in square braces, which is primarily used in shell scripts.

Files

- b *file*** *file* is a block device
- c *file*** *file* is a character device
- d *file*** *file* is a directory
- f *file*** *file* is a normal file
- g *file*** set group ID bit (SGID) of *file* is set
- G *file*** effective GID matches owner's group
- k *file*** sticky bit of *file* is set
- O *file*** effective UID matches the file owner
- p *file*** *file* is a named pipe
- r *file*** *file* exists and is readable
- s *file*** *file* is larger than 0 bytes
- S *file*** *file* is a socket
- t [*n*]** file descriptor *n* corresponds to a terminal
- u *file*** set user ID bit (SUID) of *file* is set
- w *file*** *file* exists and is writable
- x *file*** *file* exists and is executable
- d1* -ef *d2*** *files d1* and *d2* are linked
- d1* -nt *d2*** *file d1* is newer than *file d2*
- d1* -ot *d2*** *file d1* is older than *file d2*

Character strings

- n *z1*** length of character string *z1* is greater than zero
- z *z1*** length of character string *z1* is zero
- z1*** character string *z1* is not null
- z1* = *z2*** *z1* is equal to *z2*
- z1* != *z2*** *z1* is not equal to *z2*
- z1* < *z2*** *z1* is lexicographically smaller than *z2*
- z1* > *z2*** *z1* is lexicographically greater than *z2*

Numeric conditions

n1 -eq n2	<i>n1</i> equals <i>n2</i>
n1 -ge n2	<i>n1</i> is greater than or equal to <i>n2</i>
n1 -gt n2	<i>n1</i> is greater than <i>n2</i>
n1 -le n2	<i>n1</i> is less than or equal to <i>n2</i>
n1 -lt n2	<i>n1</i> is smaller than <i>n2</i>
n1 -ne n2	<i>n1</i> is not equal to <i>n2</i>

Combinatoric conditions

! a1	true if expression <i>a1</i> is false
a1 -a a2	true if <i>a1</i> and <i>a2</i> are true
a1 -o a2	true if <i>a1</i> or <i>a2</i> is true

Examples

```
if [ -f /etc/shadow ]
    tests whether file /etc/shadow exists
if [ "$res" != "j" ]
    Does the content of the variable res equal "j"?
while [ -z "$res" ]
    Does the variable res contain an empty string?
```

time command [arguments]

Executes the specified command and then displays the execution time.

time

touch [options] files

Changes the last access date/time and the last modification date/time of files. If a specified file does not exist, it is created as empty.

touch

Available options

-a	changes only the time of the last access
-c	disables the creation of empty files
-m	changes only the time of the last modification
-r file	transfers the time from a specified reference <i>file</i>
-t value	sets the file date and the system time to the specified

value with the format MMDDhhmm (month, day, hour, minute)

tr

tr [options] [string1 [string2]]

Copies the standard input device to the standard output device and in the process replaces or deletes characters. If a character from *string1* is found in the standard input device, then it is replaced with the corresponding character from *string2*.

Available options

- c** outputs the complement of the set of characters in *string1*
 - d** deletes characters that appear in *string1*
 - s** suppresses repeated sequences in the output
-

troff

troff [options] [files]

Formats files for printer or linotronic machine (also see *nroff* and *groff*)

true

true

This command returns only 0 ("successful") as return code. It is used primarily in shell scripts.

umask

umask [value]

Outputs the current value of the file generation mask as an octal number or sets this *value*. This mask determines the maximum permissions that a newly created file can receive. Here the *umask value* is subtracted from the permissions of the file to be created.

uname

uname [options]

Outputs the name and version number of the current system.

Available options

- a** outputs all available information
- m** outputs hardware (processor) type
- n** outputs the host name
- r** outputs the version number of the operating system
- s** outputs the name of the operating system
- v** outputs the date and time of compilation of the kernel

uncompress [options] [files]

Restores an original file that was compressed with `compress`.

uncompress

Available options

- c** outputs the file contents to the standard output device.
Here `uncompress` behaves like the command `zcat`

uniq [options] [file1[file2]]

Deletes successive identical lines in the linewise sorted `file1` and outputs these to `file2` (or the standard output device).

uniq

Available options

- c** outputs the number of repetitions
- d** outputs only lines that occur redundantly
- u** outputs only lines that occur uniquely
- n** skips a number *n* of fields (with tabs and blanks as delimiters) before comparing two lines
- +n** skips *n* characters before beginning to compare
- w** specifies the number of characters to be compared

uptime

Outputs the current time, the time since the last reboot, the number of logged users, and the momentary system load.

uptime

Reference

uudecode

uudecode [file]

Decodes a file encoded with `uuencode` using its original name, owner and permissions.

uuencode

uuencode [file1] name

Encodes binary files so that they can be represented as ASCII files and sent via e-mail. An encoded file is 35% larger than the original. The result is written to the standard output device. The specified name corresponds to the filename after the file is unpacked by its recipient.

vi

vi [options] [files]

Full-screen editor for processing ASCII files. It is largely based on `ex` and generally functions on all terminals.

w

w [options] [users]

Displays all currently logged users and their activities. Without parameters, all users are output; with a name, only the specified user.

Available options

- h** suppresses a title line
 - f** determines whether the login terminal should also be output
 - s** concise output format
-

wc

wc [options] [files]

Counts the number of characters, words and lines in a text file.

Available options

- c** only the number of characters
 - l** only the number of lines
 - w** only the number of words
-

Reference

whatis [commands]

Outputs a short description of the specified commands from the Online Manual.

whatis

which [commands]

Outputs the file path of the specified commands (usually an internal shell command).

which

who [options] [file]

Outputs a list of currently users currently logged in, their terminals, the login time and the name of the host on which they logged in. If a filename is specified in addition, then this file is used for evaluation instead of `/etc/utmp`.

who

Available options

- am i** outputs the user's own data
 - i** outputs how long the user was inactive
 - H** outputs column headings
 - q** outputs only the login name and the number of users
 - w** displays whether the user accepts (+) messages generated with `write` or not (-)
-

write user [terminal]

Outputs a message on a certain user's terminal. The message is read by the standard input device until it encounters EOF (<Ctrl-d>).

write

xargs [options] [commands]

Executes a command with the (multiple) arguments read from the standard input device. This enables passing lists of any length of arguments to commands

xargs

Available options

- 0** filenames are terminated by the character null

Reference

-
- e *string*** ends processing as soon as the specified character *string* appears in the list of filenames (default is "_")
 - ln** executes the command with *n* arguments
 - nn** executes the command with at most *n* arguments
 - p** interactive processing where the user must respond with "y" before a command is executed
 - s *n*** each argument may contain at most *n* characters
 - t** displays the command before its execution
-

zcat [*files*]

zcat

Decompresses the specified *files* and outputs their contents to the standard output device. The compressed files remain untouched.